



Ricardo Jorge Domingos Lares

Licenciado em Ciências da Engenharia Electrotécnica e de
Computadores

**Sistema de iluminação eficiente utilizando a
tecnologia LED para espaços públicos
interiores**

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: João Pedro Abreu Oliveira, Professor Doutor,
Faculdade de Ciências e Tecnologia da
Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutora Maria Helena Fino
Arguente: Prof. Doutor João Francisco Alves Martins
Vogal: Prof. Doutor João Pedro Abreu de Oliveira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro de 2013

Sistema de iluminação eficiente utilizando a tecnologia LED para espaços públicos interiores

Copyright © Ricardo Jorge Domingos Lares, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa tem o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor’.

Também, de acordo com os Regulamentos dos Cursos de 2.º, 3.º ciclos e Mestrados Integrados, e o Despacho 41/2010 de 21 de Dezembro de 2010, as teses sujeitas a período de embargo só são divulgadas quando este período terminar. Um período de embargo da divulgação também pode ser solicitado para as dissertações elaboradas com base em artigos previamente publicados por outros editores, sempre que tal seja necessário para respeitar os direitos de cópia desses editores.

Agradecimentos

Agradeço aos meus Pais, Irmã, Cunhado e Afilhada pelo incentivo, pelos “puxões” de orelhas, e apoio incondicional em todos os momentos.

Agradeço à minha Namorada pelo incentivo para concluir esta dissertação de mestrado, pelo apoio em momentos menos bons e pela paciência ao acompanhar todos os momentos, que nem sempre foram agradáveis no desenrolar desta dissertação.

Agradeço aos meus Amigos mais chegados pela ajuda e não “desencaminhamento” ao longo do desenrolar da dissertação de mestrado.

Agradeço ao meu orientador Prof. Doutor João Pedro Abreu Oliveira pelas ideias, apoio e disponibilidade que demonstrou. Agradeço os ensinamentos que me proporcionou ao longo do meu percurso curricular como professor de alguns módulos do curso.

Agradeço aos professores do departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa pelo incentivo e ensinamentos académicos transmitidos ao longo do meu percurso curricular.

Agradeço ao Eng. José Luís Coelho pelo apoio e dicas ao longo deste projecto, e ao longo da minha vida profissional na Empresa X.

Agradeço ao Frederico Lucas pelo apoio e ensinamentos em fotometria.

Agradeço às minhas colegas da Empresa X pelo incentivo, compreensão e ajuda na elaboração deste projecto.

A todos, os referidos ou esquecidos, um **muito obrigado**.

Resumo

A emergência de novas plataformas inteligentes e colaborativas tem impulsionado o aparecimento de novas soluções tecnológicas que irão convergir em sistemas mais sustentáveis. Para dar corpo a este novo paradigma, estas plataformas têm de suportar um número crescente de sensores e actuadores mais sofisticados, que terão de cooperar entre si de forma inteligente e eficiente.

Desde o final do século XIX que a iluminação eléctrica tem sido um dos principais factores de evolução do mundo moderno. Todavia, as soluções tradicionais de iluminação eléctrica apresentam níveis de eficiência energética reduzidos. Portanto, o desenvolvimento de sistemas de iluminação sustentáveis e colaborativos contribui para uma economia de baixo índice de carbono.

O trabalho exposto nesta Dissertação apresenta um sistema de iluminação inteligente baseado em tecnologia LED, para aplicação em espaços interiores. Para além de um cuidado dimensionamento dos circuitos electrónicos associados, é demonstrado que o estabelecimento de inter-conectividade entre pontos de luz contribui para a melhoria da eficiência energética do sistema. Esta malha colaborativa entre luminárias é obtida através da integração de um módulo de rádio ZigBee de 2,4 GHz em cada ponto de luz. Foi igualmente desenvolvido um protocolo específico e otimizado, para além da implementação de um circuito de alimentação e controlo da cadeia de LED.

O sistema de iluminação inteligente proposto foi implementado e testado com sucesso, tanto electricamente como fotometricamente. Quando comparados com sistemas de iluminação convencionais, foi possível atingir uma redução de energia até 59%, sendo este nível reforçado ao ativar-se a interligação entre luminárias.

Palavras-chave: Eficiência energética, Iluminação LED, *Driver* para LEDs, Sistema de gestão, Comunicação Zig-Bee.

Abstract

The emerging of collective awareness platforms opened a new range of driven forces that will converge to more sustainable systems. To achieve this task, these platforms have to support an increasing number and more sophisticated remote sensors and actuators that will need to cooperate smartly and strongly with each other in a mesh type of intelligent interconnectivity.

Since the late of the 19th century, electrical illumination has been one of the key factors responsible for the evolution of the modern world but it is also responsible for a significant inefficient consumption of energy. Therefore, promoting sustainable and collaborative illumination system is a basis for an effective Low-Carbon economy.

The work presents a smart LED based illumination system for indoor environments that by using a real-time mesh wireless connectivity between light spots, an high efficiency is achieved. To enable this, each spot is equipped with a low power and low duty cycled wireless ZigBee 2.4 GHz transceiver. A dedicated and optimized command protocol has also been developed. Additionally, further optimization was reached by designing a modified BUCK LED driver.

The proposed LED system has been tested, both electrically and photometrically, to evaluate the overall system performance. When compared with conventional illumination systems a power reduction up to 59% was achieved but when activating interconnectivity between light spots, the power reduction is further improved.

Keywords: *Energy efficiency, LED lighting, LED driver, Management System, Zig-Bee communication.*

Índice

AGRADECIMENTOS	V
RESUMO	VII
ABSTRACT	IX
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABELAS	XVII
LISTA DE SIGLAS UTILIZADAS	XIX
1. INTRODUÇÃO	1
1.1.MOTIVAÇÃO E ENQUADRAMENTO	1
1.2.OBJECTIVOS	2
1.3.ESTRUTURA DA DISSERTAÇÃO	2
2. CONCEITOS BÁSICOS DE ILUMINAÇÃO	5
2.1.CARACTERIZAÇÃO DA LUZ	5
2.1.1.ESPECTRO VISÍVEL E COR	6
2.1.2.COR BRANCA E CONFORTO VISUAL	9
2.2.RADIOMETRIA E FOTOMETRIA DA LUZ	11
2.3.TIPO DE ILUMINAÇÃO	14
2.3.1.LUMINÁRIAS CONVENCIONAIS DE INCANDESCÊNCIA	14
2.3.2.LUMINÁRIAS CONVENCIONAIS DE DESCARGA	15
2.3.3.LUMINÁRIAS COM DÍODO EMISSOR DE LUZ	17
2.4.COMPARAÇÃO ENTRE TIPOS DE ILUMINAÇÃO	19
3. TECNOLOGIA PARA ILUMINAÇÃO A LED	21
3.1.DÍODO EMISSOR DE LUZ (LED)	21
3.1.1.CARACTERÍSTICAS ELÉCTRICAS DO LED	22
3.1.2.EMIÇÃO DE LUZ DE COR BRANCA UTILIZANDO O LED	24
3.2.DRIVERS PARA LED	26
3.2.1.DRIVER DE TENSÃO CONSTANTE	26
3.2.2.DRIVER EM CORRENTE	27
3.3.DRIVER COM FONTE COMUTADA	33
4. TECNOLOGIAS PARA GESTÃO DE SISTEMAS DE ILUMINAÇÃO	51
4.1.ESTRATÉGIAS DE CONTROLO PARA SISTEMAS DE ILUMINAÇÃO	51
4.2.ARQUITECTURAS DE CONTROLO	55
4.3.TIPOS DE COMUNICAÇÃO NO DOMÍNIO DA ILUMINAÇÃO INTELIGENTE	59
4.4.COMUNICAÇÃO ZIG-BEE	63
5. PROJECTO DE UM SISTEMA DE ILUMINAÇÃO LED INTELIGENTE	65
5.1.DESCRICÇÃO GERAL DO SISTEMA	65
5.1.1.UNIDADE DE LEDS	65
5.1.2.DIMENSIONAMENTO DO DRIVER	68

5.2.PROJECTO DO SISTEMA DE CONTROLO	74
5.2.1.UNIDADE DE CONTROLO DA LUMINÁRIA	76
5.2.2.UNIDADE DE COMUNICAÇÃO	80
5.3.FERRAMENTA DE TESTE E PROGRAMAÇÃO	86
6. ENSAIO EXPERIMENTAL DO SISTEMA DE ILUMINAÇÃO	89
6.1.MONTAGEM DO PROTÓTIPO	89
6.1.1.PROTÓTIPO COM MÓDULO DE GESTÃO E COMUNICAÇÃO	91
6.2.TESTE ELÉCTRICO	91
6.2.1.COMPORTAMENTO DA PLACA DE LEDS	91
6.2.2.COMPORTAMENTO DO DRIVER	93
6.3.TESTE FOTOMÉTRICO	98
6.3.1.DIAGRAMAS DE RADIAÇÃO	98
6.3.2EFICIÊNCIA LUMINOSA DA LUMINÁRIA	99
6.4.AVALIAÇÃO ENERGÉTICA DE UM SISTEMA DE LUMINÁRIAS	99
6.4.1.SEM SISTEMA DE GESTÃO APLICADO	100
6.4.2.SISTEMA DE GESTÃO E COMUNICAÇÃO APLICADO	102
6.5.ANÁLISE DA AVALIAÇÃO ENERGÉTICA DOS SISTEMAS DE LUMINÁRIAS	103
7. CONCLUSÃO E TRABALHOS FUTUROS	105
7.1.CONCLUSÕES	105
7.2.TRABALHOS FUTUROS	105
BIBLIOGRAFIA	107
ANEXOS	112

Índice de Figuras

Figura 1.1 - Fontes de produção de energia eléctrica em 2012.	1
Figura 2.1 - Ondas electromagnéticas [5].....	5
Figura 2.2 - Espectro das ondas electromagnéticas, [5].	5
Figura 2.3 - Espectro visível, [7].....	6
Figura 2.4 - Curvas da sensibilidade do olho humano ao espectro visível, [5].	7
Figura 2.5 - Escala de temperaturas de cor [5].	7
Figura 2.6 - Diagrama de cromaticidade do modelo CIE [8].	8
Figura 2.7 - Palette de cores da CIE para cálculo do CRI [11].	8
Figura 2.8 - Qualificação de CRI's pela OSRAM, [9].	9
Figura 2.9 - Cor branca no modelo CIE [8].	10
Figura 2.10 - Relação de conforto visual entre iluminância e a sua Tc, [9].	10
Figura 2.11 - Diagrama polar da luminária Luminestra LED da Osram, [16].....	12
Figura 2.12 - Representação de uma superfície aparente para cálculo da luminância, [9].....	13
Figura 2.13 - Tipos de iluminação convencional.	14
Figura 2.14 - Constituintes de uma luminária LED.	17
Figura 2.15 - Aplicação de uma lente de 15°.	18
Figura 2.16 - LED SMD com dissipador aplicado.	18
Figura 3.1 - Aplicação de tensão directa, em cima. Recombinação <i>electrão-lacuna</i> , em baixo.	21
Figura 3.2 - Cone de escape de luz emitida.	22
Figura 3.3 - Curva característica de um LED.	23
Figura 3.4 - Circuito equivalente de um díodo ideal.	23
Figura 3.5 - Circuito equivalente de um díodo real.	24
Figura 3.6 - Luz branca a partir de três cores primárias.	24
Figura 3.7 - Aplicação de fósforo sobre um LED base.	25
Figura 3.8 - Comprimento de onda resultante da conversão com base em LED azul, [19].	25
Figura 3.9 - a) Curva de corrente directa em relação a Vf de LR5360, [2].b), - Curva de corrente directa em relação a Vf do OSRON SSL150, [20].	26
Figura 3.10 - a) Curva do fluxo luminoso em relação a If de LR5360 [23].	27
Figura 3.11 - Circuito de polarização de um LED em corrente.	27
Figura 3.12 - Circuito resultante para um LED LR5360 a 10mA.	28
Figura 3.13 - Circuito resultante para um LED Osron SSL 150 a 350mA.	29
Figura 3.14 - Circuito de uma fonte de corrente usando um regulador de tensão (LM317) [4].	30
Figura 3.15 - Circuito comum utilizando o BCR402R [5].	31
Figura 3.16 - Gráfico da corrente de saída vs tensão de referência, [5].	32
Figura 3.17 - Cálculo da resistência externa do BCR402R.	33
Figura 3.18 - Circuito típico de um driver <i>Buck</i> [26].	34
Figura 3.19 - Primeiro ciclo de um driver <i>Buck</i>	34
Figura 3.20 - Segundo ciclo de um driver <i>Buck</i>	35
Figura 3.21 - Forma de onda da corrente no transistor durante o primeiro ciclo na topologia <i>buck</i>	35
Figura 3.22 - Formas de onda da corrente no díodo durante o segundo ciclo na topologia <i>buck</i>	36
Figura 3.23 - Formas de onda na bobine.	36
Figura 3.24 - Formas de onda na bobine no modo CCM.	38
Figura 3.25 - Formas de onda na bobine no modo DCM.	38
Figura 3.26 - Formas de onda na bobine em modo BCM.	39
Figura 3.27 - Implementação da forma de controlo por tensão, à direita, e forma de controlo por corrente, à esquerda.	40
Figura 3.28 - Formas de onda PWM com diferentes <i>duty cycles</i>	40

Figura 3.29 - Implementação de controlo por PWM.....	41
Figura 3.30 - Representação do funcionamento em frequência variável e t_{off} constante.....	41
Figura 3.31 - Circuito típico de utilização do LT3590 [10].	42
Figura 3.32 - Formas de onda da comutação e da corrente da bobine.	42
Figura 3.33 - Circuito típico de um <i>driver Boost</i>	43
Figura 3.34 - Formas de onda no transistor durante o primeiro ciclo na topologia <i>boost</i>	43
Figura 3.35 - Formas de onda da corrente no transistor durante o primeiro ciclo na topologia <i>boost</i> . .	44
Figura 3.36 - Circulação da corrente no segundo ciclo de um <i>driver boost</i>	44
Figura 3.37 - Formas de onda no transistor durante o segundo ciclo na topologia <i>boost</i>	45
Figura 3.38 - Formas de onda da corrente resultante na topologia <i>boost</i>	45
Figura 3.39 - Formas de onda da corrente resultante na topologia <i>boost</i>	45
Figura 3.40 - Circuito típico de um <i>driver Flyback</i>	46
Figura 3.41 - Andamento da corrente durante o primeiro ciclo na topologia <i>Flyback</i>	46
Figura 3.42 - Andamento da corrente durante o segundo ciclo na topologia <i>Flyback</i>	47
Figura 3.43 - Perdas de condução e comutação de um MOS-FET, [11].	48
Figura 3.44 - Perdas de condução e comutação de um diodo, [11].	49
Figura 3.45 - Tetraedro de potências.	49
Figura 3.46 - Circuito Valley-Fill.....	50
Figura 3.47 - Formas de onda à saída do circuito <i>Valley-Fill</i> , [12].	50
Figura 4.1 - Algoritmo de decisão dependendo do valor de luminosidade.	51
Figura 4.2 - Exemplo de aplicação do algoritmo de decisão dependendo do valor de luminosidade. ...	52
Figura 4.3 - Sistema de gestão com sensores de luminosidade locais.....	52
Figura 4.4 - Sistema de gestão com estimação de ocupação com base em programação horária.	53
Figura 4.5 - Sistema de gestão com controlo de ocupação real com base em uso de sensores de presença.	54
Figura 4.6 - Sistema de gestão com estratégia composta com base em uso de sensores de presença e de luminosidade.	54
Figura 4.7 - Arquitectura de um sistema de gestão isolada.	56
Figura 4.8 - Exemplo de uso da arquitectura de um sistema de gestão autónoma.	56
Figura 4.9 - Exemplo de uso da arquitectura de um sistema de gestão zona a zona sem gestão centralizada.....	57
Figura 4.10 - Exemplo de uso da arquitectura de um sistema de gestão zona a zona com gestão centralizada.....	57
Figura 4.11 - Exemplo de uso da arquitectura de um sistema de gestão centralizada.....	58
Figura 4.12 - Exemplo de uso da arquitectura de um sistema de gestão composta.....	59
Figura 4.13 - Exemplo de interligação de um sistema com protocolo DMX512.	60
Figura 4.14 - Estrutura de um pacote DMX512.	60
Figura 4.15 - Estrutura da arquitectura Stand Alone de um sistema DALI.....	61
Figura 4.16 - Estrutura da arquitectura de um subsistema <i>Stand-Alone</i> de um sistema DALI integrado no sistema de gestão do edifício.....	61
Figura 4.17 - Estrutura da arquitectura de um sistema DALI fazendo parte integrante do sistema de gestão do edifício.	61
Figura 4.18 - Esquema de ligações de um sistema DALI.	62
Figura 4.19 - Estrutura de um pacote DALI.....	62
Figura 4.20 - Frequências e canais disponíveis para Zig-Bee.	63
Figura 5.1 - Esquema da placa de LEDs.	65
Figura 5.2 - Gráfico da relação entre a corrente e o fluxo luminoso a 350 mA [51].	66
Figura 5.3 - Circuito típico usando o SFH3710.	67
Figura 5.4 - Circuito típico usando o LM335.....	67
Figura 5.5 - Circuitos típicos usando o HV9910b.....	69
Figura 5.6 - Circuito do driver resultante do dimensionamento.....	74
Figura 5.7 - Exemplo de uma situação estado possível das luminárias em um espaço.....	75

Figura 5.8 - Placa de desenvolvimento Arduino UNO.....	76
Figura 5.9 - Esquema de ligação para interacção entre luminárias.	76
Figura 5.10 - Esquema de ligação da saída de PWM ao driver.....	76
Figura 5.11 - Diagrama de funcionamento normal da luminária proposta.....	78
Figura 5.12 - Estrutura do vector das luminárias circundantes.	79
Figura 5.13 - Módulo XBee Series 2 da Digi®.	80
Figura 5.14 - Configuração da rede Zig-Bee no <i>software</i> X-CTU.....	81
Figura 5.15 - Estrutura da trama definida.....	81
Figura 5.16 - Exemplos de tramas.....	82
Figura 5.17 - Diagrama de recepção de dados através do módulo de comunicação.	83
Figura 5.18 - Diagrama de envio de dados através do módulo de comunicação.....	84
Figura 5.19 - Esquema de ligação para a ferramenta programação e teste.	86
Figura 5.20 - Interface para configuração da porta série.....	86
Figura 5.21 - Interface para leitura/gestão da luminária.....	87
Figura 5.22 - Interface para gestão das luminárias circundantes.....	88
Figura 6.1 - Esquemático final da placa de LEDs do protótipo.	89
Figura 6.2 - Circuito impresso final da placa de LEDs do protótipo. (<i>Top layer</i> a vermelho e <i>bottom layer</i> a azul).....	89
Figura 6.3 - Circuito final da placa de driver.	90
Figura 6.4 - Circuito impresso final da placa de driver do protótipo. (<i>Top layer</i> a vermelho e <i>bottom layer</i> a azul).....	90
Figura 6.5 - Circuito impresso final da placa de driver do protótipo todo montado.	91
Figura 6.6 - Temperatura vs Horas de vida útil, [6].	92
Figura 6.7 - Forma de onda do pino CS do HV9910b. (Escala: Tensão:100mV/Div. Tempo:5us/Div.)	93
Figura 6.8 - Forma de onda da gate do transístor. (Escala: Tensão:2V/Div. Tempo:5us/Div.).....	94
Figura 6.9 - Forma de onda da gate do transístor. (Escala: Tensão:100mV/Div. Tempo:5us/Div.)....	94
Figura 6.10 - Esquema de ligações para medição da corrente dos LED's.	95
Figura 6.11 - Forma de onda da corrente fornecida aos LEDs. (Escala: Tensão:100mV/Div. Tempo:5us/Div.)	95
Figura 6.12 - Forma de onda da corrente com PWM a 100%. (Escala: Tensão:100mV/Div. Tempo:500us/Div.)	96
Figura 6.13 - Forma de onda da corrente com PWM a 75%. (Escala: Tensão:100mV/Div. Tempo:500us/Div.)	96
Figura 6.14 - Forma de onda da corrente com PWM a 50%. (Escala: Tensão:100mV/Div. Tempo:500us/Div.)	97
Figura 6.15 - Diagrama polar e cartesiano fotométrico do protótipo da luminária proposta.	99
Figura 6.16 - Distribuição das luminárias convencionais e representação das linhas isográficas de iluminância.....	100
Figura 6.17 - Distribuição das luminárias protótipo a LED e representação das linhas isográficas de iluminância.....	101
Figura 6.18 - Distribuição das luminárias protótipo a LED e representação das linhas isográficas de iluminância.....	101
Figura 6.19 - Cenário de utilização da sala com luminárias protótipo e sistema de gestão e comunicação implementados.	102

Índice de Tabelas

Tabela 2.1 - Espectro electromagnético, [4].....	6
Tabela 2.2 - Tabela de grandezas radiométricas e fotométricas, [10].	11
Tabela 2.3 - Comparação entre fontes de luz convencionais	19
Tabela 3.1 – Valores para o circuito com o LED LR5360, tendo em conta a tolerância da resistência e a variação de V_f	28
Tabela 3.2 - Valores para o circuito com o LED Oslon SSL 150, tendo em conta a tolerância da resistência e da variação de V_f	29
Tabela 3.3 – Valores para o circuito com o LED Oslon SSL 150 e com o LM317 tendo em conta a tolerância da resistência.....	31
Tabela 4.1 - Valores de luminância segundo a norma DIN 5035-2:1990, [9].....	52
Tabela 4.2 - Comparação entre estratégias de gestão.....	55
Tabela 5.1 - Especificações de dimensionamento do driver	68
Tabela 5.2 - Exemplo de valores de consumo com interacção entre luminárias.....	75
Tabela 5.3 - Modos de funcionamento da luminária.	77
Tabela 5.4 - Tipos e funções dos códigos das tramas.....	85
Tabela 6.1 - Temperaturas dos LEDs aplicados na placa com dissipador.....	92
Tabela 6.2 - Resultados dos testes efectuados ao driver a 100%.	97
Tabela 6.3 - Valores resultantes do cenário proposto com a luminária protótipo completa.	103

Lista de siglas utilizadas

A	Ampère
API	<i>Application Programming Interface</i>
BCM	<i>Boundary Conduction Mode</i>
CCM	<i>Continuous Conduction Mode</i>
cd	Candela
CDL	Curva de distribuição luminosa
CFL	<i>Compact Fluorescent Lamp</i>
CIE	<i>Commision Internationale l'Eclairage</i>
CQS	<i>Color Quality Scale</i>
CRI	<i>Color Rendering Index</i>
CSMA-CA	<i>Carrier Sense Multiple Access Collision Avoidence</i>
DCM	<i>Discontinuous Conduction Mode</i>
Dsss	<i>Direct Sequence Spread Spectrum</i>
Ev	Iluminância
eV	<i>electronVolt</i>
Iv	Intensidade Luminosa
J	<i>Joule</i>
LED	<i>Light-Emmiting Diode</i>
lm	<i>Lúmen</i>
lm/m²	<i>Lúmen por metro quadrado</i>
lm/W	<i>Lúmen por Watt</i>
Lv	Luminância
O-QPSK	<i>Offset Quadrature Phase Shift Keying</i>
PAN	<i>Personal Area Network</i>
PWM	<i>Pulse Width Modulation</i>
Rdson	Resistência de condução de um transístor de efeito de campo
s	Segundos
Tc	Temperatura de cor
UV	Ultra-Violeta
V	<i>Volt</i>
Vf	Tensão directa do díodo
W	<i>Watt</i>
ZC	Zig-Bee Coordinator
ZCL	Zig-Bee Cluster
ZED	Zig-Bee End-Device
ZR	Zig-Bee Router
Φv	Fluxo Luminoso
Ω	<i>Ohm</i>

1. Introdução

1.1. Motivação e Enquadramento

O crescimento continuado do consumo de energia eléctrica e a necessidade de se tornar mais sustentável todo processo de produção, tem-se traduzido na ativação de meios alternativos de geração de eletricidade. Por exemplo, em 2012 a EDP anunciou, [1], que cerca de 40% da energia produzida é de origem eólica, Figura 1.1.

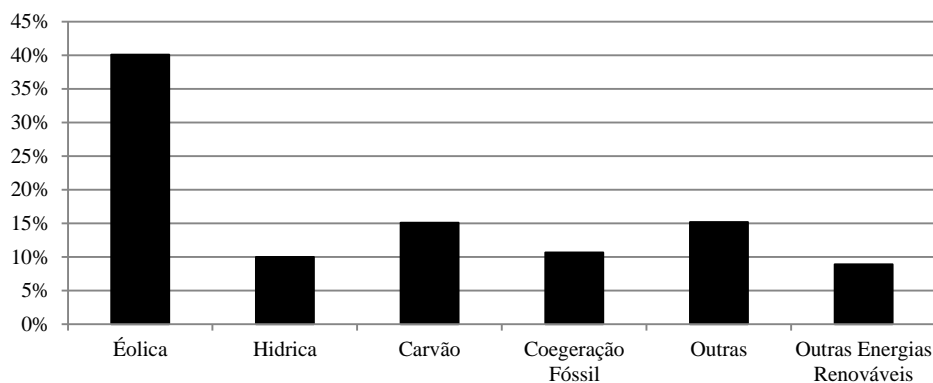


Figura 1.1 - Fontes de produção de energia eléctrica em 2012.

Todavia, também o perfil do consumo deverá atingir níveis de sustentabilidade compatíveis com os recursos existentes. É neste enquadramento que a eficiência energética passou a constituir uma importante variável de projeto, nomeadamente, ao nível da electrificação de edifícios. Com esta crescente preocupação da eficiência energética de um edifício, resultado da definição e atribuição de certificações oficiais, novas estratégias foram sendo exigidas e aplicadas. Dentro deste conjunto, tem sido dada atenção especial ao consumo energético na componente de iluminação, o qual ascende a 28% do total consumido num edifício [2].

Com o desenvolvimento tecnológico têm surgido várias soluções de iluminação pública mais eficientes, com a consequente diminuição acentuada do uso de iluminação incandescente tradicional. Esta foi sendo substituída por luminárias de descarga. Verifica-se atualmente, a emergência de iluminação com base na tecnologia LED justificada pelos elevados níveis de eficiência energética que esta apresenta. Dada a facilidade de controlo deste tipo de dispositivo, os diversos fabricantes têm vindo a adicionar novos produtos que permitem aumentar os níveis de eficiência, passando não só pela introdução de modificações no processo de emissão de fótons com também pela utilização de módulos de gestão inteligente para optimizarem o uso da iluminação. Alguns destes sistemas (fonte de luz e sistema de gestão) atingem níveis de reduções de consumo que poderão ultrapassar os 40%. Um exemplo real da aplicação de passagem de iluminação convencional em halogénio para iluminação a LED, ocorrido numa instalação hoteleira em território nacional, resultou numa poupança efetiva de até

75%. Em termos concretos existiam na unidade cerca de 440 unidades halogéneo com 60 W cada, com período de funcionamento diário de 16 horas, representando um consumo anual de 177.197 kWh (15.071€ a custos de 2008). Tendo estas luminárias sido substituídas por lâmpadas de LED com 17W, o consumo anual reduziu-se para 44.197 kWh, o que representou uma poupança de 11.314€ (considerando custos de 2009), [3].

Para além desta poupança a nível energético teremos de considerar que a vida útil de uma luminária LED é bastante superior, quando comparado com iluminação convencional. Esta pode atingir as 100.000 horas.

Com estas características e com um baixo custo produtivo, temos uma solução bastante competitiva para substituição de quaisquer sistemas de luminárias que utilizem iluminação convencional.

1.2. Objectivos

Esta dissertação tem como objectivo principal apresentar um sistema inteligente para gestão de luminárias com tecnologia LED, de modo a atingir elevados níveis de eficiência energética.

Pretende-se que o sistema LED proposto seja mais eficiente do que os sistemas convencionais e para isso terá em conta não só o desenvolvimento do sistema de gestão per si mas também o dimensionamento electrónico do *driver* e da própria fonte de luz.

Este sistema inclui um sistema de gestão e de comunicação que aumentará o nível de “inteligência” de modo a dispor de autonomia de decisão e de interação com as restantes luminárias existentes no espaço interior.

Com este sistema prevê-se poupanças energéticas bastante acentuadas tanto a nível comparativo com iluminação convencional, como com iluminação a LED sem colaboração com luminárias circundantes.

1.3. Estrutura da dissertação

A presente dissertação está estruturada em diversos capítulos, para além deste capítulo de introdução ao tema.

No capítulo 2 serão dados conceitos de luminotécnica, bem como informações relevantes sobre iluminação convencional e iluminação a LED. É igualmente efectuada a análise comparativa entre as soluções de iluminação apresentadas.

No capítulo 3 será descrito o dispositivo LED, não só ao nível eléctrico, como ao nível óptico. Será feito um estudo do melhor tipo de *driver* LED a utilizar. Neste capítulo, são apresentadas as estratégias e arquiteturas de controlo possíveis em um sistema inteligente de iluminação.

No capítulo 4, são apresentadas arquiteturas e estratégias de controlo de iluminação.

No capítulo 5, é apresentado o dimensionamento da luminária LED, tanto a placa de LED bem como o respectivo *driver*. É igualmente apresentada uma solução de um sistema inteligente de gestão da

luminária bem como um sistema de comunicação para permitir a interação entre as luminárias com vista à otimização da gestão das luminárias.

No capítulo 6 é feita uma avaliação do sistema em termos de comportamentos eléctricos, comportamentos fotométricos, e sendo demonstrada a eficiência energética do protótipo da luminária proposta. É feita também uma análise comparativa entre as prestações da luminária protótipo com luminárias convencionais.

No capítulo 7, são enumeradas as principais conclusões, sendo apresentado possibilidades de trabalho futuro para optimização do sistema aqui proposto.

2. Conceitos básicos de iluminação

Para dar suporte à análise e projecto de uma luminária, descrevem-se neste capítulo alguns dos principais conceitos e características associadas à luminotécnica.

2.1. Caracterização da luz

No campo da radiação electromagnética [4], representada na Figura 2.1 pelos vectores do campo eléctrico (E) e magnético (B), a luz situa-se numa zona do espectro que é visível ao olho humano. Por outro lado, atendendo ao princípio da dualidade onda-partícula, para além de poder ser descrita por uma equação de onda, o quanta da emissão de luz é o fóton. Esta é a partícula elementar das radiações electromagnéticas que apesar de representar um quanta de energia, não tem massa. Esta partícula viaja à máxima velocidade cujo o valor é $C = 300 \times 10^6$ m/s , em espaço vazio.

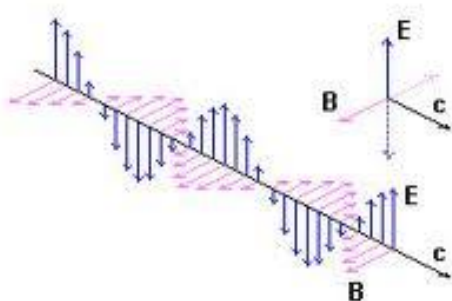


Figura 2.1 - Ondas electromagnéticas [5].

Esta velocidade também representa a velocidade propagação do campo electromagnético e permite relacionar o comprimento de onda λ (em metros) e a frequência f (em Hertz) da onda através da Equação 2.1 .

$$\lambda = \frac{C}{f} [m] \quad (2.1)$$

A distribuição espectral de comprimentos de onda está representada na Figura 2.2 e detalhada na Tabela 2.1.

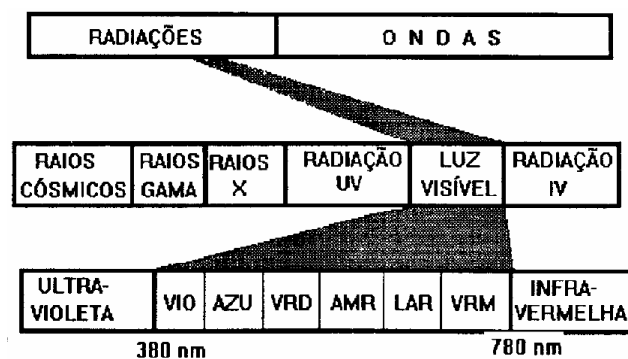


Figura 2.2 - Espectro das ondas electromagnéticas, [5].

Espectro electromagnético		
Faixa	Frequências	Comprimentos de onda
Áudio	10 a 20KHz	15000000 a 15000 m
RF	10MHz a 300000MHz	30000 m a 0,1 cm
Infravermelho	10^6 a $3,9 \times 10^8$ MHz	0,03 a $7,6 \times 10^{-5}$ cm
Espectro visível	$3,9 \times 10^8$ a $7,9 \times 10^8$ MHz	$7,6 \times 10^{-5}$ a $3,8 \times 10^{-5}$ cm
Ultravioleta	$7,9 \times 10^8$ a $2,3 \times 10^{10}$ MHz	$3,8 \times 10^{-5}$ a $1,3 \times 10^{-6}$ cm
Raios X	$2,3 \times 10^{10}$ a $3,0 \times 10^{13}$ MHz	$1,5 \times 10^{-5}$ a $1,0 \times 10^{-9}$ cm
Raios Gama	$2,3 \times 10^{12}$ a $3,0 \times 10^{14}$ MHz	$1,3 \times 10^{-8}$ a $1,0 \times 10^{-10}$ cm
Raios Cósmicos	$>4,8 \times 10^{14}$ MHz	$<6,25 \times 10^{11}$ cm

Tabela 2.1 - Espectro electromagnético, [4].

2.1.1. Espectro Visível e Cor

O espectro visível é a região do espectro das ondas electromagnéticas com comprimento de onda compreendido entre os 380 nm e os 780 nm. Dentro desta faixa, a cada comprimento de onda está associada uma cor, cuja distribuição espectral está representada na Figura 2.3 .

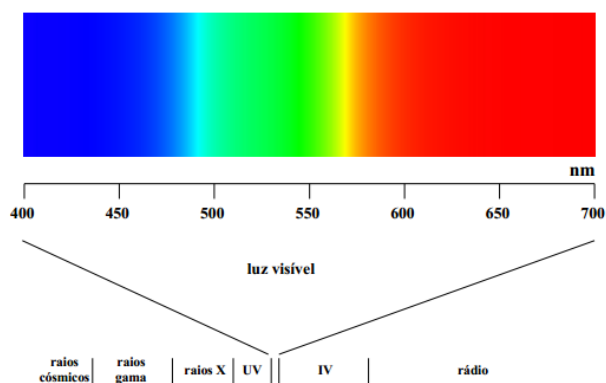


Figura 2.3 - Espectro visível, [7].

Pela figura podemos observar, que a cor violeta está entre os 380 nm e os 450 nm. A cor azul está entre 450nm e os 490nm. A cor verde está entre os 490 nm e os 570 nm. A cor amarela este entre os 570 nm e os 590 nm. A cor laranja está entre os 590 nm e os 620 nm e por fim a cor vermelha está entre os 620 nm e os 780 nm. Todas as restantes cores podem ser obtidas pela mistura de duas ou mais cores, [7].

O olho humano apresenta uma sensibilidade à cor que se altera em função das condições de iluminação, conforme se ilustra na Figura 2.4. Por exemplo, para condições de mais baixa luminosidade (à noite), a visão humana é mais sensível ao verde e azul.

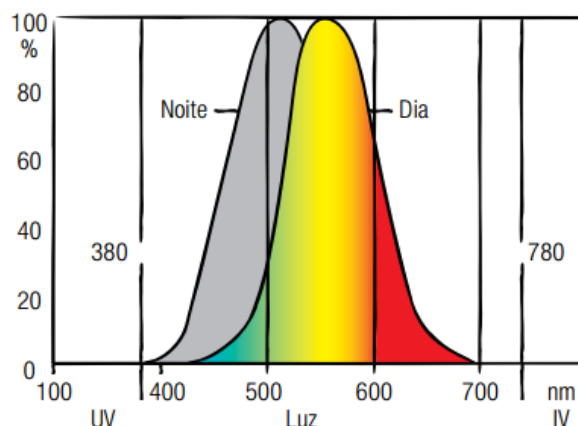


Figura 2.4 - Curvas da sensibilidade do olho humano ao espectro visível, [5].

Temperatura de cor (T_c)

Uma forma alternativa de caracterizar a cor é pela sua temperatura de cor, sendo este valor de utilização habitual em projectos de iluminação (em particular, para luz branca). A temperatura de cor de uma fonte de luz é definida pelo seu calor ou pela sua frieza aparente, e é medida em graus Kelvin [8]. A Figura 2.5 mostra a escala das temperaturas de cor existente.

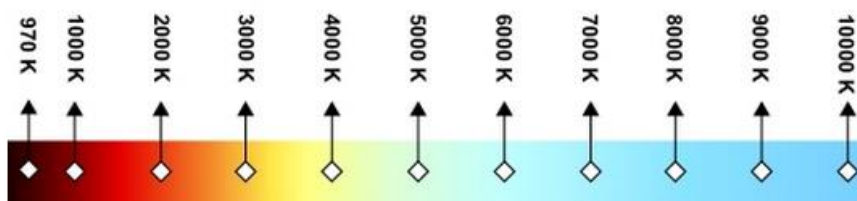


Figura 2.5 - Escala de temperaturas de cor [5].

Esta temperatura de cor é medida através de um corpo negro que, quando aquecido, vai mudando de cor consoante a temperatura atingida [9]. A Figura 2.5 revela que cores quentes apresentam temperatura de cor mais baixas.

Modelo CIE

A *Commision Internationale l'Éclairaige* (CIE) criou em 1931 um modelo para representar todas as cores do espectro visível, [7]. A aplicação do modelo resulta no diagrama de cromaticidade que é apresentado na Figura 2.6.

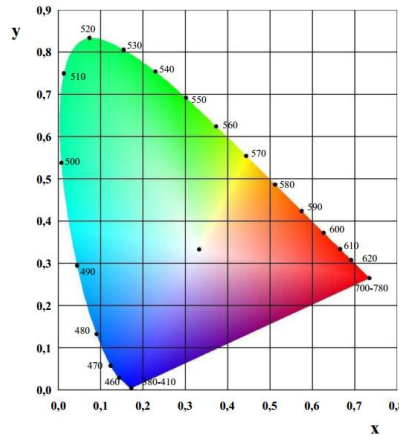


Figura 2.6 - Diagrama de cromaticidade do modelo CIE [8].

Neste diagrama, a cor é representada através de um ponto com coordenadas C_x e C_y . Uma propriedade importante deste modelo é a sua linearidade [7].

Color Rendering Index

Para se aferir o impacto de uma fonte de luz na percepção da cor de um objecto, define-se o indicador *Color Rendering Index* (CRI). A tradução de CRI para português é índice de reprodução de cores (IRC ou R_a). O índice CRI traduz o desvio da cor de um objecto iluminado por uma fonte de luz relativamente ao caso de iluminação por uma fonte de luz de referência, [10]. Esta última é designada por “*Planckian Radiator*” se a temperatura for inferior a 5000 K, e se a T_c for superior a 5000 K a referência é designada por “*CIE Daylight source*”, [11].

A determinação o índice é efectuada com base na iluminação de uma paleta de cores previamente definida pela CIE (representada na Figura 2.7).



Figura 2.7 - Paleta de cores da CIE para cálculo do CRI [11].

O cálculo índice CRI é efectuada através da média obtida para as oito cores da paleta, [11],

$$CRI = \frac{1}{8} \sum_{i=1}^8 R_i \quad (2.2)$$

em que R_i corresponde ao IRC individual. Por sua vez, R_i é dado por

$$R_i = 100 - 4.6\Delta E_i \quad (2.3)$$

em que ΔE_i corresponde ao desvio relativamente ao caso de iluminação por uma fonte referência.

Podendo variar entre 0 a 100 (sendo este o melhor valor), esta escala permite classificar a fonte de luz em função do seu impacto na percepção da cor e assim contextualizar a sua instalação. A figura 2.8 define quatro níveis de percepção indicados por alguns fabricantes, [9].

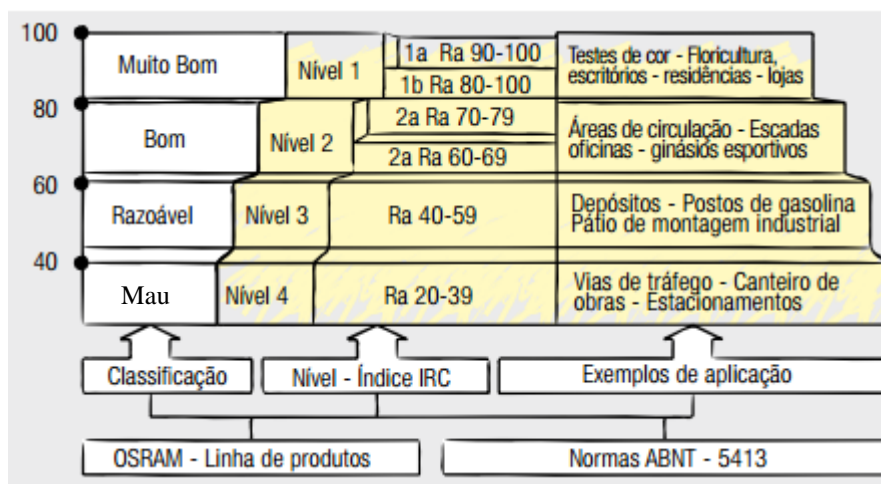


Figura 2.8 - Qualificação de CRI's pela OSRAM, [9].

No entanto esta medida de CRI apresenta alguns problemas quando aplicados a fontes de luz proveniente de LEDs. Dado que nenhuma das oito cores de referência usadas não são completamente saturadas. Isto representa um problema especialmente para os LED de cor branca, [11]. A saturação de cores pode ser reduzida apesar de apresentar um nível de CRI elevado. Outro problema, é o facto de uma fonte de luz puder gerar um espectro de luz com qualidade aceitável nas oito cores de referência mas não nas restantes, [11]. Para ultrapassar isso foi definido o conceito de escala de qualidade de cor (CQS). Para fazer a quantificação da qualidade de cor utilizou-se uma paleta de 15 cores em vez das 8 referidas anteriormente. O método de obter o valor de CQS é idêntico ao do CRI. Com isto consegue-se precisar com maior exactidão a distribuição da qualidade de cores de uma fonte de luz, [12].

2.1.2. Cor Branca e Conforto Visual

A cor branca, com Tc entre 2700 K e os 6500 K, localiza-se na região definida pelas coordenadas $0,310 \leq C_y \leq 0,410$ e $0,310 \leq C_x \leq 0,410$, no modelo CIE. A Figura 2.9 ilustra a localização, no diagrama de cromaticidade, de uma luz branca com Tc compreendida entre os 3500 K e os 6000 K.

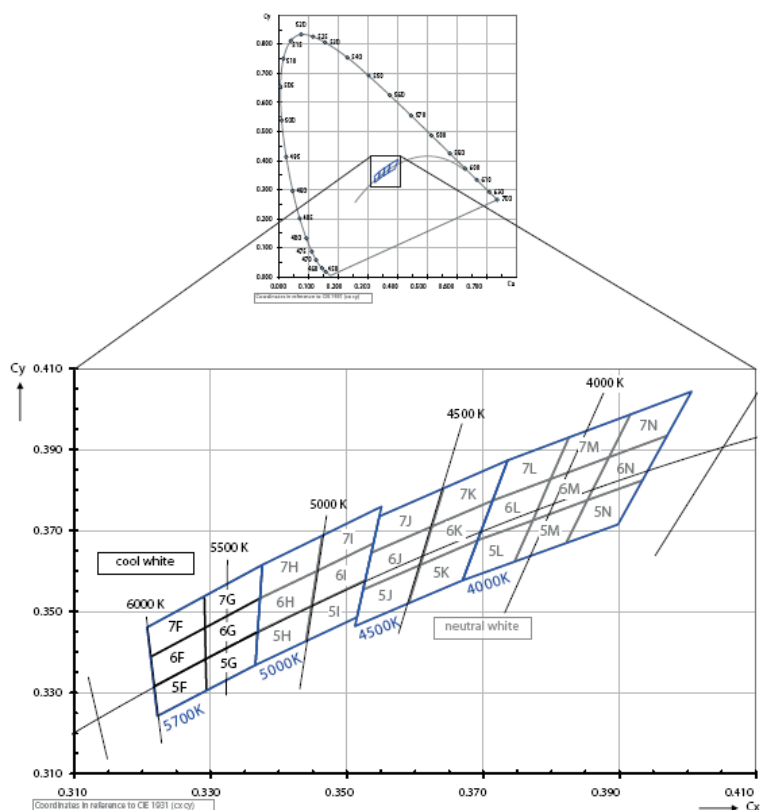


Figura 2.9 – Cor branca no modelo CIE [8].

A cor de uma fonte de luz pode provocar alterações no conforto visual, tendo em conta o número de horas de exposição, [9]. Dados de um fabricante (OSRAM) exemplifica uma zona de conforto (diagrama de Kruithof), representada na Figura 2.10.

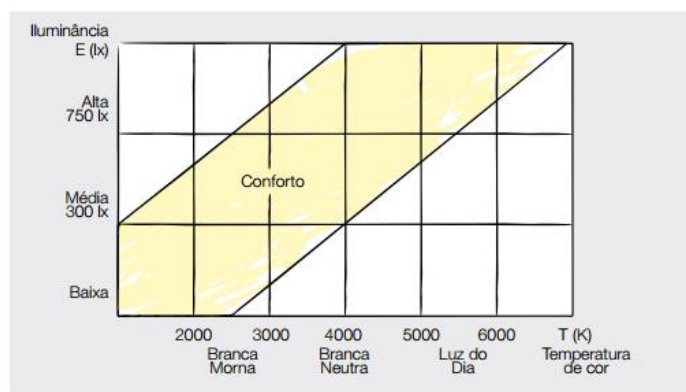


Figura 2.10 - Relação de conforto visual entre iluminância e a sua Tc, [9].

A zona de conforto indica que quanto maior for a iluminância mais fria deverá a cor, ou seja, maior terá de ser a sua Tc. Por exemplo, uma fonte luminosa com cor mais fria desperta a atenção do ser humano e, por isso, é preferencialmente utilizada em locais de trabalho e lojas. Por sua vez, uma fonte luminosa com uma temperatura de cor mais quente reforça uma sensação de conforto, sendo utilizada em habitações e em locais de descanso, [9].

2.2. Radiometria e fotometria da luz

A radiometria descreve a detecção e medida da radiação electromagnética pelo espectro de frequências. Por sua vez, a fotometria pretende quantificar a resposta espectral da sensibilidade do olho humano, [13]. A tabela 2.2 ilustra a relação entre as grandezas em causa.

Radiometria			Fotometria		
Quantidade	Símbolo	Unidade	Quantidade	Símbolo	Unidade
Fluxo de radiação	Φ_e	[W]	Fluxo luminoso	Φ_v	[lm]
Intensidade de radiação	I_e	[W/sr]	Intensidade luminosa	I_v	[cd]
Irradiância	E_e	[W/m ²]	Iluminância	E_v	[lm/m ²]
Radiância	L_e	[W/m ² -sr]	Luminância	L_v	[lm/m ² -sr]

Tabela 2.2 - Tabela de grandezas radiométricas e fotométricas, [10].

Fluxo luminoso (Φ_v)

Define-se por fluxo luminoso a quantidade total de radiação que atravessa uma superfície, emitida por uma fonte de luz, por unidade de tempo à qual o olho humano é sensível. A unidade do fluxo luminoso é o lúmen (lm), [7], podendo ser calculada através de,

$$\Phi_v = \int_s (\bar{I} \cdot \bar{n}) ds \quad [lm] \quad (2.4)$$

A medição prática desta grandeza é feita recorrendo a uma esfera de *Ulbricht*. Este equipamento mede a quantidade total de luz em lúmens por comparação a uma fonte de luz calibrada, [14].

Intensidade luminosa (I_v)

A intensidade luminosa I_v é o fluxo luminoso irradiado por ângulo sólido Ω , numa dada direcção. A unidade da intensidade luminoso é o candela (cd), [7], podendo ser calculada por,

$$I = \frac{\Phi_v}{\Omega} [cd] \quad (2.5)$$

onde Φ_v é o fluxo luminoso [lm] e Ω é o Ângulo sólido em radiano.

A representação gráfica da intensidade luminosa, designada por curva de distribuição luminosa (CDL), fornece informação sobre a distribuição fotométrica da luminária. Esta distribuição angular é traçada no plano transversal (C0/180) e longitudinal (C90/270) da luminária, possibilitando assim prever a distribuição espacial de luz emitida pela luminária, [7]. Na Figura 2.11, é ilustrado um exemplo de uma distribuição fotométrica referente ao modelo da OSRAM Luminestra LED, [16].

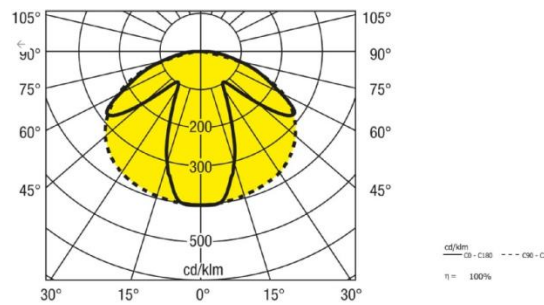


Figura 2.11 – Diagrama polar da luminária Luminestra LED da Osram, [16].

Iluminância

Define-se iluminância, [7] e [15], como o fluxo luminoso irradiado por uma fonte de luz que incide numa determinada área A , medido em lúmen por metro quadrado [lm/m^2], ou seja em lux [lx], podendo ser definida através de,

$$E_v = \frac{\Phi_v}{A} \quad \left[\frac{\text{lm}}{\text{m}^2} \right] \quad (2.6)$$

onde Φ_v é o fluxo luminoso [lm] e A a área irradiada [m^2].

A iluminância poderá ser calculada depende directamente da intensidade luminosa e da distância à fonte de luz. Para afastamentos superiores a 5 vezes o tamanho da fonte, a iluminância varia com o inverso do quadrado da distância, [5],

$$E_v = \frac{I}{d^2} \quad [\text{lx}] \quad (2.7)$$

onde, I é intensidade luminosa em candela e d é a distância à fonte de luz em metro.

Esta é uma grandeza fundamental no cálculo luminotécnico, visto que permite fornecer o valor da densidade de fluxo luminoso nas superfícies desejadas, por exemplo, numa mesa de escritório.

Luminância

Define-se luminância a razão entre a intensidade luminosa emitida pela fonte luz, em relação à superfície aparente, [9]. A superfície aparente corresponde à área perpendicular ao campo de visão humana, conforme representado na Figura 2.12 .

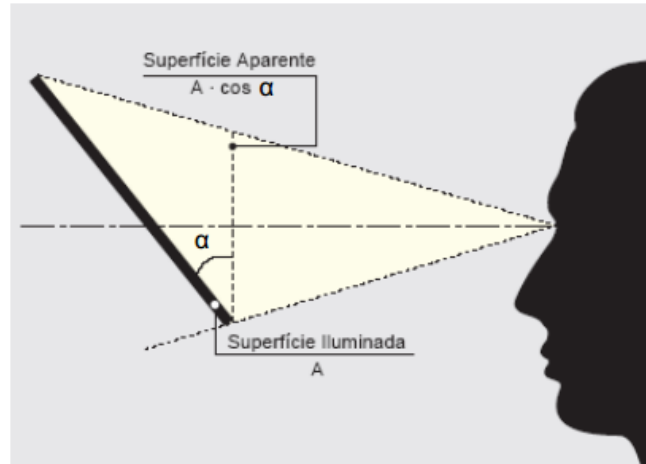


Figura 2.12 - Representação de uma superfície aparente para cálculo da luminância, [9].

A luminância corresponda à intensidade luminosa associada à área aparente, [9], podendo se obtida pela expressão dada por,

$$L = \frac{I}{A \cdot \cos(\alpha)} \quad [cd/m^2] \quad (2.8)$$

onde I é a intensidade luminosa [cd] e $A \cdot \cos(\alpha)$ é a área aparente $[m^2]$. Alternativamente, a expressão poderá ser dada em função do coeficiente de reflexão,

$$L = \frac{\rho \cdot E}{\pi} \quad [cd/m^2], \quad (2.9)$$

onde ρ é o coeficiente de reflexão e E é a Iluminância $[lm/m^2]$.

Eficiência

A eficiência de uma fonte de luz traduz a relação entre fluxo luminoso e a potência eléctrica que esta consome. Por outras palavras, a eficiência representa a quantidade de luz por unidade de potência eléctrica, [3,6]. Esta eficiência é calculada através da expressão dada por,

$$\eta = \frac{\Phi_v}{P} \quad [lm/W], \quad (2.10)$$

onde Φ_v é o fluxo luminoso [lm] e P é potência eléctrica associada à fonte de luz [W].

A título exemplificativo, apresentam-se dois modelos de lâmpadas, com níveis de eficiência distintos. A primeira corresponde ao modelo OSRAM CLASSIC A 40 W 230 V E27 que dispõe um fluxo luminoso de 415 lm para uma potência de 40W, [14]. A sua eficiência é de 10,80 lm/W. A segunda corresponde ao modelo OSRAM L 15 W/840 que tem um fluxo luminoso de 950 lm e uma potência de 15 W [16]. A sua eficiência é de 63,3 lm/W.

2.3. Tipo de iluminação

A tipificação luminotécnica é feita em termos do funcionamento da fonte de luz, construção, aplicação alvo e, naturalmente, em termos dos rendimentos atingidos. A tecnologia LED apresenta rendimentos elevados. Contudo, o parque instalado com iluminação convencional é extenso e variado, conforme se pode observar na Figura 2.13.

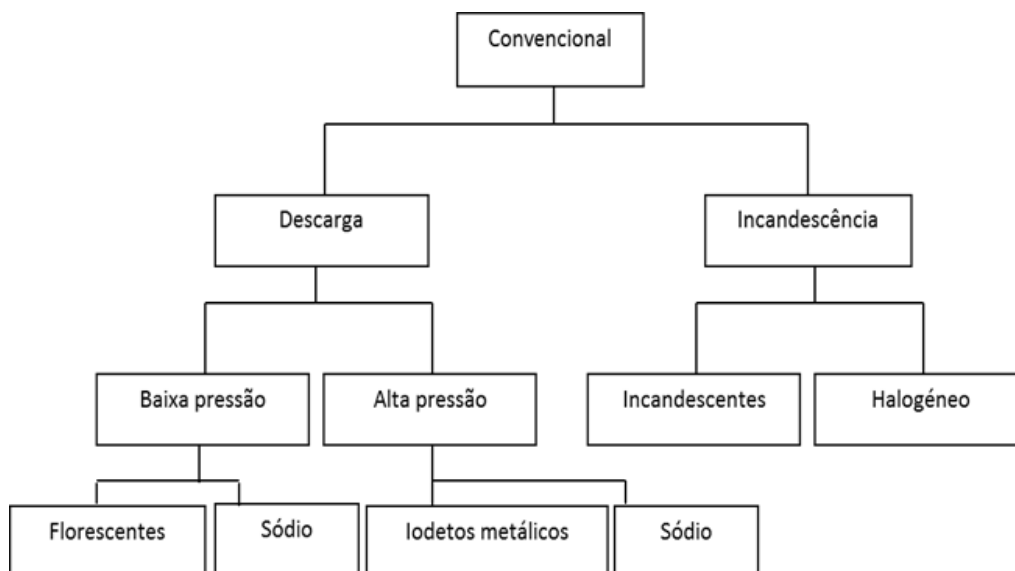


Figura 2.13 - Tipos de iluminação convencional.

2.3.1. Luminárias convencionais de Incandescência

Lâmpadas incandescentes

Este tipo de lâmpadas é constituído por um invólucro de vidro que tem no seu interior um filamento de tungsténio e por um gás inerte. O funcionamento consiste em fazer percorrer uma corrente eléctrica através do filamento, o qual vai aquecer e emitir luz. Esta emissão de fotões não é eficiente porque apenas cerca de 5 a 10% da energia aplicada é transformada em luz, sendo que a restante é termicamente radiada.

Este tipo de lâmpadas têm uma temperatura de cor que ronda os 2700 K, apresentam um CRI de 100 e a sua vida útil ronda as 1000 horas de funcionamento. O tempo de passagem de 0% a 100% do fluxo é muito reduzido, [17]. A eficiência deste tipo de lâmpadas é baixa, podendo variar entre os 10 e os 15 lm/W, [15].

Lâmpadas de halogéneo

A constituição das lâmpadas de halogéneo é bastante semelhante às anteriores, com a diferença no tipo gás utilizado, que pode ser iodo ou bromo. O seu funcionamento consiste em fazer percorrer uma corrente eléctrica no filamento, que provoca temperaturas na ordem dos 3000 K o que faz libertar tungsténio evaporado, [17]. A interação deste com o gás halogéneo é devido aos mecanismos de

convexão, as partículas voltam a depositar-se no filamento, dando origem a novo ciclo. Isto prolonga a durabilidade do filamento e evita o escurecimento provocado por tungsténio evaporado. Esta lâmpada tem dimensões mais reduzidas que uma de incandescência, podendo de até 100 vezes menor, [17].

Este tipo de lâmpada tem uma temperatura de cor que ronda os 3000 K e os 3200 K, apresenta um CRI de aproximadamente 100 e a sua vida útil varia entre 2000 e as 4000 horas. O tempo de passagem de 0% a 100% do fluxo é praticamente nulo, [17].

A eficiência deste tipo de lâmpadas é superior ao das lâmpadas incandescentes, mas ainda assim é bastante reduzido visto que ainda dissipa muita da energia fornecida em forma de calor. O seu rendimento ronda os 15 lm/W e os 25 lm/W, [17].

Por último, este tipo de lâmpadas emite mais radiação ultravioleta que as incandescentes, mas os níveis emitidos são inferiores à radiação solar, logo não apresenta riscos para a saúde pública, [17].

2.3.2. Luminárias Convencionais de Descarga

As lâmpadas de descarga apresentam melhores desempenhos que as de incandescência. O seu funcionamento consiste em provocar ionização e posterior libertação de radiação. Isto é conseguido pelo aparecimento de alta tensão nos filamentos que libertam alguns electrões para o meio gasoso inerte. Esta libertação provoca movimento de electrões que ao chocarem com os átomos do gás tendem a libertar mais electrões. Com isto formam-se pares electrões/íões, originando um efeito de avalanche, cuja ionização faz com que seja libertada radiação UV. É esta radiação UV que ao atravessar um material, como por exemplo fósforo, produz ondas electromagnéticas no espectro da luz visível, [17]. Para o funcionamento das lâmpadas de descarga é necessária a aplicação de um balastro, podendo este ser electrónico ou ferromagnético, consoante o tipo de fluorescente em utilização.

Lâmpadas de descarga de baixa pressão

Lâmpadas fluorescentes tubulares

Na sua constituição é utilizado o gás árgon a baixa pressão e o tubo é coberto com um material à base de fósforo para converter a radiação UV em radiação visível. Se o gás utilizado for vapor de mercúrio, tensão mais elevadas devem ser aplicadas aos eléctrodos, [17]. Este tipo de lâmpada pode ser aplicado em diversos ambientes como por exemplo escritórios, residências e superfícies comerciais.

Estas lâmpadas têm uma temperatura de cor que ronda entre os 3000 K e os 8000 K, apresenta um CRI entre 75 e os 95 e a sua vida útil pode ir das 12000 horas até às 75000 horas. O tempo de passagem de 0% a 100% do fluxo é um processo não instantâneo podendo atingir algumas dezenas de segundos, [17]. A eficiência deste tipo de lâmpadas ronda os 60 lm/W e pode ultrapassar os 80 lm/W [16].

Fluorescentes Compactas

Estas lâmpadas são as de uso residencial mais comum no que diz respeito a lâmpadas economizadoras. São apelidadas de CFL (*Compact Fluorescent Lamp*). A sua constituição é semelhante às tubulares com a diferença de o tubo de descarga ter dimensões mais reduzidas. Outra diferença é a utilização de tri-fósforo no revestimento do tubo, formado por três compostos com uma banda estreita e centrados no comprimento de onda do azul, vermelho e verde. Com a combinação apropriada destas cores, é possível alterar a temperatura de cor da luz branca.

Este tipo de lâmpada tem uma temperatura de cor que ronda os 3000 K e os 6500 K, apresenta um CRI entre 80 e os 89 e a sua vida útil pode ir das 6000 horas até as 20000 horas, [16]. O tempo de passagem de 0% a 100% do fluxo é um processo que não é instantâneo, o tempo varia com a lâmpada mas é menor que 1 minuto, [17].

A eficiência deste tipo de lâmpadas varia entre 55 lm/W e os 80lm/W, [16].

Lâmpadas de sódio

Na sua constituição é utilizado gás do tipo vapor de sódio a baixa pressão, o que permite reduzir os efeitos de auto absorção, [17].

Este tipo de lâmpada tem uma temperatura de cor que ronda os 2000 K, apresenta um CRI inferior a 25, o mais baixo da iluminação convencional. Isto faz com que este tipo de lâmpada seja apenas utilizada em ruas com pouco tráfego pedestre, túneis e autoestradas. A sua vida útil pode ir das 12000 horas até as 18000 horas em condições especiais [16].

A passagem de 0% a 100% do fluxo é um processo lento, podendo atingir os 5 minutos, [17].

A eficiência deste tipo de lâmpadas é extremamente elevada podendo ultrapassar os 180 lm/W, [16].

Lâmpadas de descarga de alta pressão

Lâmpadas de iodetos metálicos

A sua constituição tem vapor de mercúrio em alta pressão, com mistura de emissores iónicos e emissores moleculares. Emissores iónicos podem ser iodetos de sódio, tálio ou índio, [17].

Estas lâmpadas têm uma temperatura de cor entre os 3500 K e os 7200 K, apresenta um CRI entre 80 e 95. Este tipo de lâmpada apresenta potências disponíveis bastante elevadas podendo atingir 18000 W. São utilizadas em iluminação urbana, campos de desportos, iluminação de fachadas. A sua vida útil pode ir das 3000 horas até as 9000, [16].

O tempo de passagem de 0% a 100% do fluxo é um processo lento, podendo atingir 10 minutos, [17].

A eficiência deste tipo de lâmpadas é elevada podendo ultrapassar com facilidade os 80 lm/W, [16].

2.3.3. Luminárias com Díodo Emissor de Luz

Existem vários tipos de luminária que usam como base o Díodo Emissor de Luz (LED). Estas podem ser de sinalização, arquitetural ou de iluminação. A Figura 2.14 apresenta o diagrama de blocos de uma luminária LED.

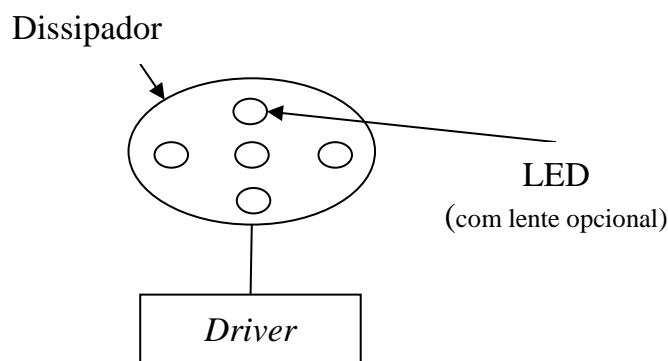


Figura 2.14 - Constituintes de uma luminária LED.

LED

A selecção do tipo do LED, de sinalização ou alto brilho, depende fortemente da aplicação alvo. Os LEDs podem emitir luz com diferentes cores, dependendo do material que foi utilizado na sua construção. A constituição do LED e o seu funcionamento será tratado no capítulo seguinte.

Óptica secundária

Em algumas aplicações é necessário direccionar a luz ou tornar o tipo de luz mais agradável. Para isso recorre-se a ópticas secundárias que podem ser lentes ou simples difusores. Todavia, ao colocar-se um material à frente de uma fonte de luz, reduz-se o seu fluxo, podendo chegar aos 20%.

A utilização de lentes permite fechar o ângulo de emissão do fluxo luminoso do LED. A título de exemplo, o estreitamento do ângulo de feixe de 80° para 15°, implica a utilização de uma lente de 15° (por exemplo, uma lente LEDIL TINA2-RS). A Figura 2.15 ilustra o efeito da lente.

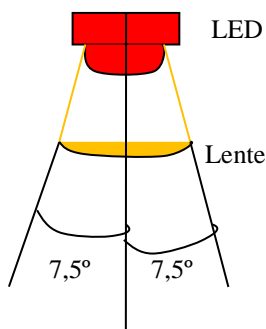


Figura 2.15 - Aplicação de uma lente de 15°.

Relativamente aos difusores, estes podem ser simétricos ou assimétricos e apresentarem diferentes níveis de translucidez ou opalescência. A utilização de difusores opalinos é justificada sempre que se pretenda suprimir o efeito de *glare*, ou seja, o efeito ofuscante do LED.

Driver

O controlo do funcionamento do LED pode ser realizado em tensão ou em corrente. Esta corrente ou tensão são geradas por um circuito dedicado à função, designado por “*driver*”, o qual será detalhadamente analisado no capítulo 3.

Gestão térmica e vida útil

Em especial no caso de utilização de LED com encapsulamento SMD (*surface mounted device*), é importante um dimensionamento térmico adequado de modo a maximizar o tempo de vida útil do dispositivo. A aplicação de um dissipador, com formato exemplificado na figura 2.16, é muita vezes inevitável. No capítulo seguinte este assunto será aprofundado.

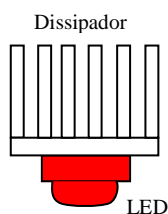


Figura 2.16 - LED SMD com dissipador aplicado.

Características típicas da luminária LED

Este tipo de luminária poderá emitir em diversas cores, dependendo do tipo e agregação de LED utilizada. No caso do LED branco, a luminária pode apresentar um CRI entre 70 e 95.

A sua vida útil, depende da gestão térmica, podendo variar entre as 25000 horas e as 120000 horas [16]. O tempo de passagem de 0% a 100% do fluxo é um processo que é instantâneo, [17].

A eficiência deste tipo de lâmpadas é elevado podendo ultrapassar os 75 lm/W [16].

2.4. Comparação entre tipos de iluminação

A Tabela 2.3 apresenta um quadro comparativo entre os diversos tipo de iluminação.

Tipo	Tc [K]	CRI	Eficiência [lm/W]	Vida útil [horas]	Tempo de 0% a 100%	Locais favoráveis de aplicação
Incandescentes	2700	100	10 a 15	1000	Instantâneo	Residências
Halogéneo	3000-3200	100	15-20	2000-4000	Instantâneo	Resistências, superfícies comerciais.
Tubulares	3000-8000	75-95	60-80	12000-75000	<1 min.	Escritórios, superfícies comerciais, etc.
CFL	3000-6500	80-89	55-80	6000-20000	<1 min.	Residências
Sódio	2000	<25	>180	12000-18000	2-5 min.	Auto-estradas, túneis, etc.
Iodetos metálicos	3500-7200	80-95	>80	3000-9000	5-10 min.	Iluminação urbana, campos de desportos
LED	Todas	70-95	>75	25000-120000	Instantâneo	Todo o tipo de aplicações

Tabela 2.3 - Comparação entre fontes de luz convencionais

Analisando a tabela anterior, observamos que a iluminação a LED tem vantagens relativamente a outras tecnologias de iluminação. Ao longo da dissertação esta demonstração ficará mais evidente.

3. Tecnologia para iluminação a LED

A iluminação convencional apresenta níveis de eficiência baixos devido à sua constituição e modo de funcionamento. Existem, no entanto, processos alternativos de emissão de luz potencialmente mais eficientes, sendo a electroluminescência um dos exemplos, [18]. Este fenómeno físico consiste na emissão de luz num material quando este é percorrido por uma corrente eléctrica ou quando este se encontra na presença de um forte campo eléctrico [19]. O processo emerge à temperatura ambiente enquanto que o fenómeno de incandescência só surge a temperaturas acima de 750°C, [18]. O primeiro díodo emissor de luz (Light Emitter Diode – LED) de cor vermelha foi inventado Holonyak e Bevacqua em 1962, [18].

3.1. Díodo Emissor de Luz (LED)

O LED é um dispositivo semicondutor de dimensões reduzidas e caracteriza-se por ser fisicamente robusto, ser estável e apresenta uma boa relação entre fluxo luminoso e potência eléctrica consumida, a qual ascende a valores acima de 100 lm/W. Sendo um dispositivo monocromático, o comprimento de onda associado à sua cor está relacionado com o tipo de material usado na sua construção, [18]. A composição AlGaInP é utilizada para fabricar LEDs de cor vermelha, amarela e alguns tons de verde. Por sua vez, o AlInGaN permite fabricar LED entre o azul e os restantes tons de verde.

A aplicação de uma tensão directa numa estrutura semicondutora $p-n$ promove a recombinação entre portadores de carga negativa e as lacunas em redor da junção, estreitando significativamente a zona de depleção, [19]. Esta recombinação consiste, na realidade, na passagem de electrões da banda de condução para a banda de valência e com isto é libertada energia na forma de fotões de luz, como se pode observar pela figura 3.1.

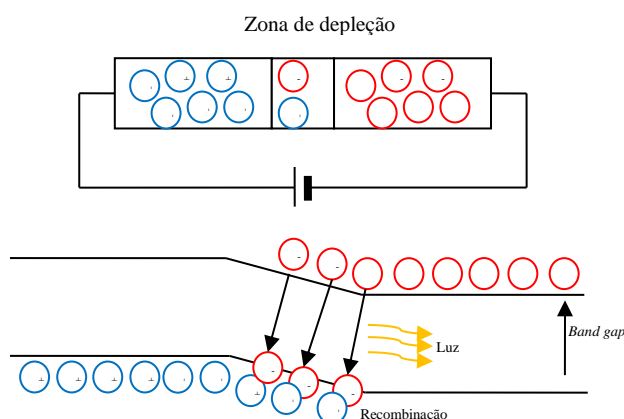


Figura 3.1 - Aplicação de tensão directa, em cima. Recombinação *electrão-lacuna*, em baixo.

A quantidade de energia libertada determina o cor e a intensidade de luz, dependendo do tipo de material constituinte da junção e do potencial de *band gap*. O *band gap* representa a diferença de energia entre duas bandas (valência e condução), sendo dada habitualmente em eV (electronvolt). A energia do fotão libertado é dada por,

$$E = hf \text{ [eV]} \quad (3.1)$$

onde h é a constante de Planck e f a frequência em hertz. A energia do fóton libertado é proporcional à sua frequência, [19].

O diagrama de radiação da luz a partir de um semicondutor é determinado pelo cone de escape que é colocado junto à zona de transição ar-semicondutor. Para isso tem-se em consideração os ângulos de reflexão internos, sendo que todos os fótons emitidos num ângulo menor que o ângulo de reflexão interna crítica (α_c), são encaminhados. A Figura 3.2 ilustra um exemplo deste processo.

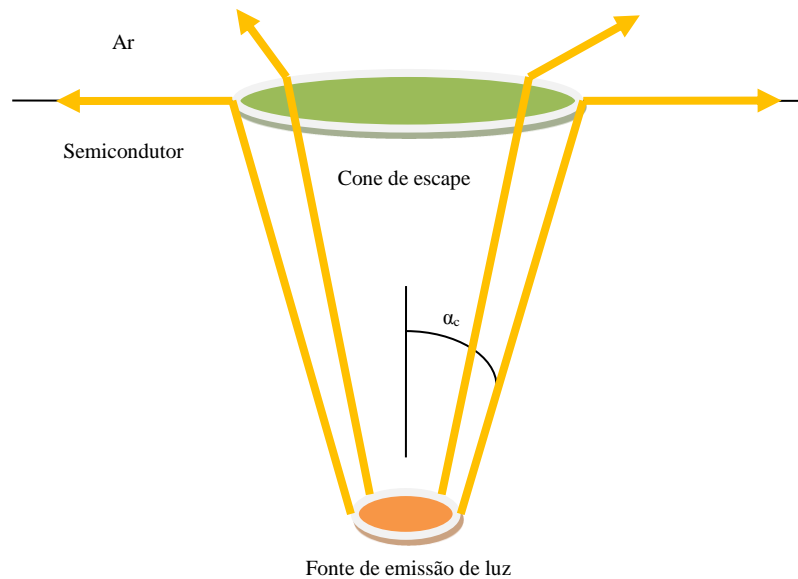


Figura 3.2 - Cone de escape de luz emitida.

Os índices de refração dos semicondutores atingem valores elevados, o que se reflete em ângulos de reflexão interna críticos pequenos. Como exemplo, o índice de refração do *GaAs* é de 3,4, [19]. A emissão de luz pode ser otimizada através do uso de encapsulamentos de *epoxy*, visto possuir um grande índice de refração, [19]. Assim, o semicondutor fica em contacto com a *epoxy* e não com o ar. O uso deste tipo de encapsulamento permite também que o ângulo de saída do fluxo luminoso seja ajustado.

3.1.1. Características eléctricas do LED

As características eléctricas de um LED são semelhantes às de um díodo normal. É um componente não linear dado a relação tensão-corrente que o caracteriza, [20]. A Figura 3.3 ilustra uma característica generalista deste tipo de dispositivo semiconductor.

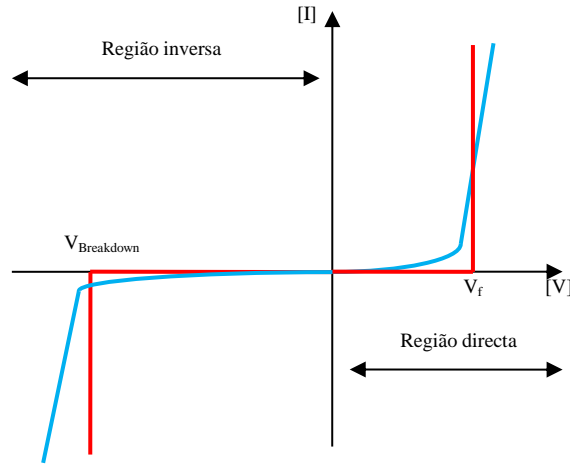


Figura 3.3 - Curva característica de um LED.

O funcionamento ideal de um LED baseia-se num modelo linear por troços, [20]. O circuito da Figura 3.4 mostra o circuito equivalente deste modelo.



Figura 3.4 - Circuito equivalente de um díodo ideal.

Utilizando este modelo ideal, a corrente do díodo é determinada simplesmente através de

$$I_D = \frac{V - V_f}{R} \quad (3.2)$$

onde V_f corresponde à tensão de condução directa do dispositivo.

O característica real é do tipo exponencial, pelo que a expressão que permite calcular a corrente é dada por,

$$I_D \approx I_{ss} \times e^{\frac{V_d}{n \times V_T}} \quad (3.3)$$

onde I_{ss} é a corrente inversa de saturação, V_d é a tensão aos terminais do díodo e V_T é a tensão térmica.

Um deles é a resistência incremental que pode ser calculada da seguinte forma, [20].

$$r_D = \frac{n \times V_T}{I_D} \quad (3.4)$$

Outro elemento parasita é a sua capacidade. Esta capacidade pode ser dividida em duas, capacidade de difusão e a capacidade de depleção, [20].

Com isto o circuito equivalente de um díodo real será o da figura 3.5.

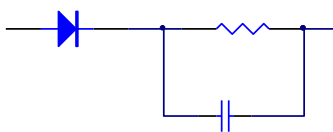


Figura 3.5 - Circuito equivalente de um díodo real.

Devido à variação de tensão relativamente à corrente como visto no ponto anterior, a forma ideal de fornecer energia a um LED é provocar uma corrente constante aos seus terminais.

Uma outra forma de fornecer energia a um LED, é aplicar um sinal de PWM (*Pulse Width Modulation*). Com a variação do *duty cycle* do PWM conseguimos regular a corrente média fornecida ao LED. Esta é uma forma muito usada na iluminação para fazer o *dimming*, que é o controlo de brilho de um LED.

Características térmicas

O aumento excessivo da temperatura da junção do LED provoca variações no fluxo luminoso e diminui a duração da vida útil do dispositivo, razão pela qual é utilizado dissipadores apropriados. A temperatura de funcionamento de um LED é poderá ser calculada a partir, [21], de

$$T_j = T_{ps} + (R_{th} \times V_f \times I_f) \quad (3.5)$$

onde T_j é a temperatura da junção [°C], T_{ps} é a temperatura no ponto de solda do LED, R_{th} é a resistência térmica entre a junção e o ponto de solda do LED, V_f é a tensão directa do LED e I_f é corrente que o atravessa.

3.1.2. Emissão de luz de cor branca utilizando o LED

Existem várias formas de gerar luz branca usando LEDs, descritas seguidamente.

Por combinação de três cores

Uma forma de gerar luz branca consiste na utilização de três LEDs distintos associados a cores primárias, designadamente o azul (440nm), o verde (525nm) e o vermelho (605nm), [19], conforme se ilustra na Figura 3.6.

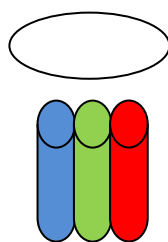


Figura 3.6 - Luz branca a partir de três cores primárias.

Esta estrutura permite atingir níveis de CRI elevados apesar de apresentar elevada sensibilidade à temperatura. Esta sensibilidade condiciona o fluxo luminoso podendo, por isso, originar desvios na temperatura de cor, [19].

Conversão do comprimento de onda

Uma forma alternativa de obter a cor branca baseia-se na conversão do comprimento de onda emitido pelo LED com cor base. A conversão é obtida pela acção de fósforo de modo a obter fotões com um segundo comprimento de onda apropriado para que, quando misturados com os gerados pela base, formem luz branca. A Figura 3.7 mostra como isso pode ser atingido.

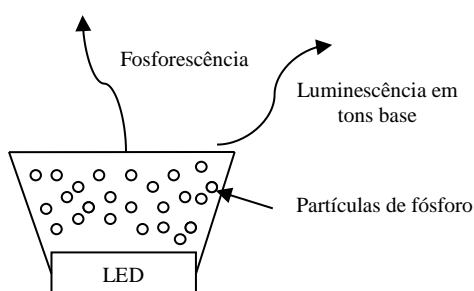


Figura 3.7 - Aplicação de fósforo sobre um LED base.

O LED base poderá ser de cor azul (GaInN/GaN). A Figura 3.8 apresenta a densidade espectral de potência óptica da luz emitida por um LED azul ao qual foi aplicado um conversor de comprimento de onda para o amarelo, como por exemplo fósforo.

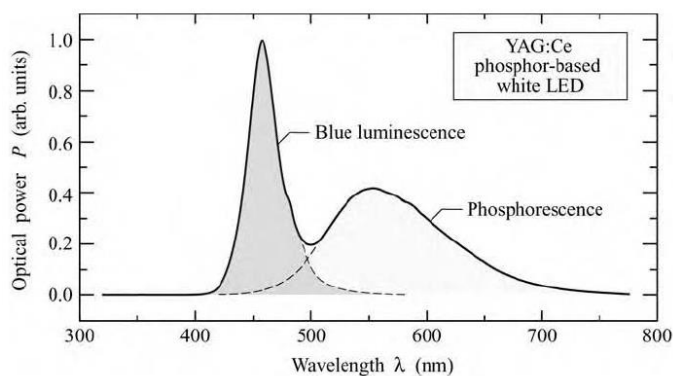


Figura 3.8 - Comprimento de onda resultante da conversão com base em LED azul, [19].

A diversidade da temperatura de cor bem como de *CRI's* é obtida através do ajuste da densidade do fósforo. Este tipo de conversão é a mais usada nos LEDs disponíveis no mercado, dado o seu elevado desempenho.

3.2. Drivers para LED

Existem essencialmente 3 formas para alimentar uma cadeia de LEDs, designadamente através de uma fonte de tensão constante, através de uma fonte de corrente constante ou utilizando uma fonte comutada com a qual se controla a corrente média de operação.

3.2.1. Driver de tensão constante

Ao aplicar uma tensão constante directamente ao LED verifica-se que a corrente que o atravessa depende de V_f e da resistência de condução do LED. A título de exemplo, analisam-se dois LEDs comerciais da OSRAM, nomeadamente um LED de sinalização LR5360, [23], e um LED de iluminação LCW CRDP.EC (Oslon SSL 150), [20]. Analisando a curva de resposta do primeiro LED (LR5360), representada na figura 3.10 a), verifica-se que a corrente deste varia significativamente para V_f entre 1,4V e 1,9V.

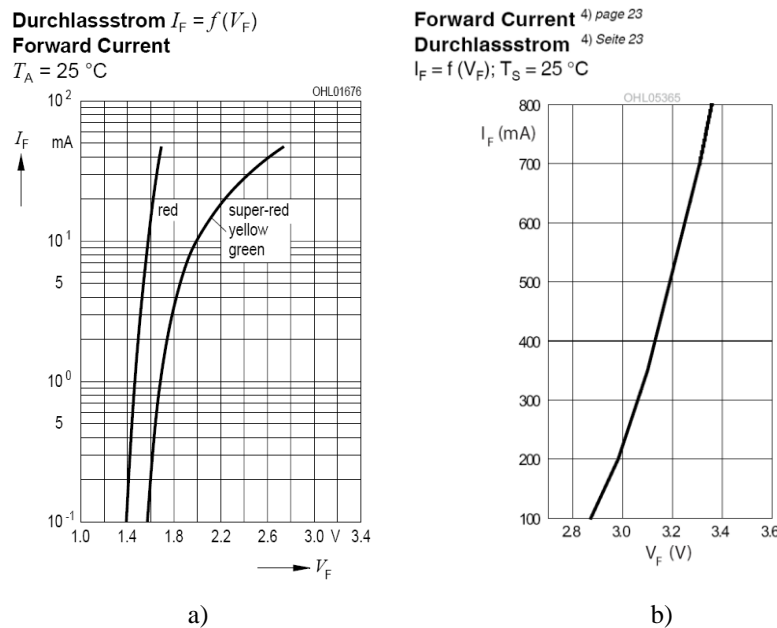


Figura 3.9 - a) Curva de corrente directa em relação a V_f de LR5360, [2]. b), - Curva de corrente directa em relação a V_f do OSLOM SSL150, [20].

Naturalmente que o nível de corrente determina o fluxo, conforme se ilustra na figura 3.10.

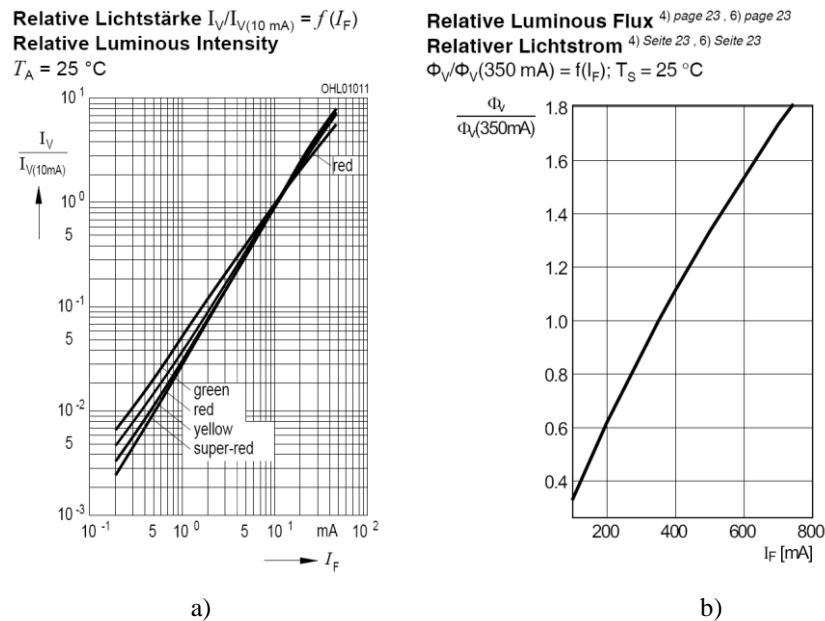


Figura 3.10 – a) Curva do fluxo luminoso em relação a I_f de LR5360 [23].

b) Curva do fluxo luminoso em relação a I_f do OSLOM SSL150, [20].

Para o segundo LED, observa-se que para uma tensão V_f igual a 3.1 V a corrente é de 350 mA. Contudo, uma variação de apenas 100 mV a corrente no diodo aumenta significativamente para 500 mA originando uma variação de fluxo luminoso de 30 % [23].

Conclui-se que o controlo do LED por via de uma fonte tensão apresenta dificuldade o controlo de fluxo luminoso.

3.2.2. Driver em corrente

A Figura 3.11 apresenta um circuito básico de polarização do LED em corrente.

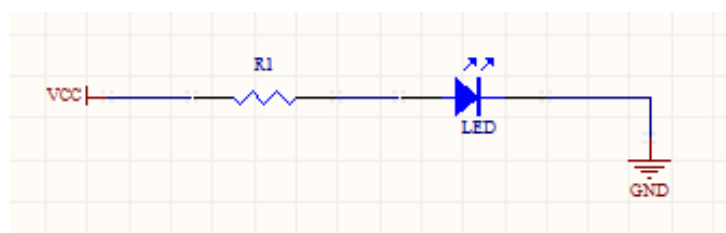


Figura 3.11 - Circuito de polarização de um LED em corrente.

Analisando o circuito, verifica-se que a corrente de polarização directa é dada por

$$I = \frac{(V_{cc} - V_f)}{R_1} \quad (3.6)$$

onde V_{cc} é a tensão de alimentação, V_f é a queda de tensão aos terminais do LED e R é o valor da resistência. Esta última é determinada com base na corrente correspondente ao fluxo luminoso pretendido e à tensão de alimentação. Considerando uma corrente de 10 mA, o LED OSRAM LR5360 (bastante comum em painéis de sinalização) desenvolve uma tensão de 1,4 V aos seus terminais, o que remete para a escolha de uma resistência de 1060Ω para uma tensão de alimentação de 12 V. O circuito resultante está representado na figura 3.12.

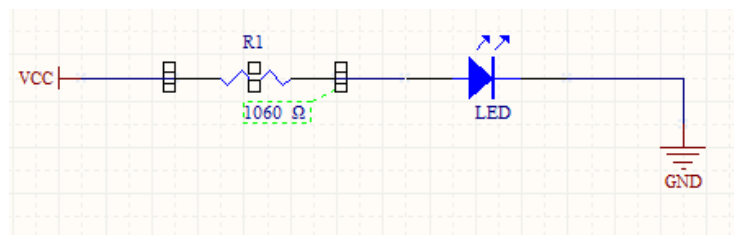


Figura 3.12 - Circuito resultante para um LED LR5360 a 10mA.

Interessa realizar uma análise de sensibilidade do circuito tendo em conta o valor da tolerância de 5% da resistência de 1060Ω , isto é, $R \pm 5\%$. A Tabela 3.1, apresenta resultados para diversos valores de V_f .

V_f [V]	I (R-5%) [mA]	I (R+5%) [mA]	Potência Resistência (R-5%) [W]	Potência Resistência (R+5%) [W]	Potência LED (R-5%) [W]	Potência LED (R+5%) [W]	Potência Total (R-5%) [W]	Potência Total (R+5%) [W]
1,40	10,53	9,52	0,1116	0,1010	0,0147	0,0133	0,1263	0,1143
1,50	10,43	9,43	0,1095	0,0991	0,0156	0,0142	0,1251	0,1132
1,60	10,33	9,34	0,1074	0,0972	0,0165	0,0150	0,1239	0,1121
1,70	10,23	9,25	0,1054	0,0953	0,0174	0,0157	0,1227	0,1111
1,80	10,13	9,16	0,1033	0,0935	0,0182	0,0165	0,1215	0,1100
1,90	10,03	9,07	0,1013	0,0917	0,0191	0,0172	0,1204	0,1089

Tabela 3.1 – Valores para o circuito com o LED LR5360, tendo em conta a tolerância da resistência e a variação de V_f .

Observa-se na Tabela 3.1 que o circuito apresentado é uma boa solução para LEDs de sinalização visto que, apesar da tolerância da resistência, não se verifica uma acentuada variação de corrente. Logo, como a variação de corrente não é muito acentuada não se obtém uma variação significativa de fluxo luminoso. Em termos de dissipação Obtém-se, no pior caso, um valor máximo de potência dissipada na resistência de 0,1116 W.

No caso do segundo LED, ou seja o Oslon SSL 80, [23], verifica-se que a situação se altera. De facto, mantendo a fonte de tensão de 12V, temos que o valor da resistência para o caso típico será de,

$$R = \frac{(V - V_f)}{I} = \frac{12 - 3,1}{0,350} = 25,43 \Omega \quad (3.7)$$

em V_f é a tensão directa do díodo para uma corrente de 350 mA. O circuito resultante está representado na Figura 3.13.

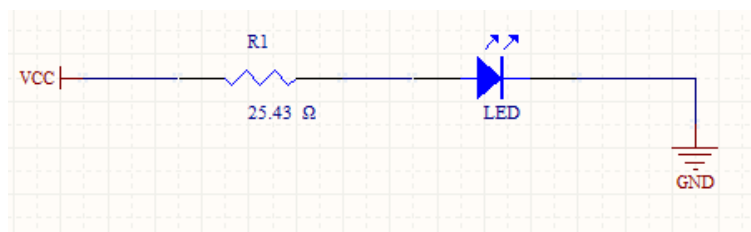


Figura 3.13 - Circuito resultante para um LED Oslon SSL 150 a 350mA.

De igual forma, interessa realizar uma análise de sensibilidade do circuito tendo em conta o valor da tolerância de 5% da resistência de 25,43Ω, isto é, $R \pm 5\%$. A Tabela 3.2, apresenta resultados para diversos valores de V_f .

V_f [V]	I (R-5%) [mA]	I (R+5%) [mA]	Potência Resistência (R-5%) [W]	Potência Resistência (R+5%) [W]	Potência LED (R-5%) [W]	Potência LED (R+5%) [W]	Potência Total (R-5%) [W]	Potência Total (R+5%) [W]
2,80	380,84	344,57	3,50	3,17	1,07	0,96	4,57	4,13
2,90	376,70	340,82	3,43	3,10	1,09	0,99	4,52	4,09
3,00	372,56	337,08	3,35	3,03	1,12	1,01	4,47	4,04
3,10	368,42	333,33	3,28	2,97	1,14	1,03	4,42	4,00
3,20	364,28	329,59	3,21	2,90	1,17	1,05	4,37	3,96

Tabela 3.2 - Valores para o circuito com o LED Oslon SSL 150, tendo em conta a tolerância da resistência e da variação de V_f .

Observa-se na Tabela 3.2 que o circuito da Figura 3.14 não é o mais indicado para este LED. A variação do valor da resistência provoca uma variação de corrente que tem como valor máximo 380,84 mA e como valor mínimo 327,72 mA. A correspondente variação de 53,12 mA, reflecte uma variação significativa do fluxo luminoso. Em termos de intensidade luminosa isto pode representar uma variação, no pior caso, de 11,2 lm se o fluxo do LED for 112 lm a 350 mA, e uma variação de 14 lm se o fluxo do LED for 140 lm, (para o modelo LCW CRDP.EC-LPLR-5H7I-1, [23]). Esta variação pode ser significativa, e ser perceptível ao olho humano se este método for aplicado numa luminária que tenha vários LEDs desta categoria.

Outra desvantagem da utilização desta topologia de circuito, é a potência dissipada na resistência. Verifica-se que o valor máximo da potência dissipada na resistência é no máximo de 3,5 W e o valor mínimo é de 2,87 W, o que revela um desperdício de energia degradando a sua eficácia energética. No pior caso, a fonte debita um valor máximo de 4,57 W, sendo 3,50 W dissipados na resistência e somente 1,07 W sendo utilizado pelo LED. Isto traduz-se numa eficiência de 23,33%, o que é bastante baixo.

Regulador de tensão usado como fonte de corrente

Uma das formas de implementar uma fonte de corrente é utilizar um regulador de tensão, como por exemplo o LM317, [24]. Este regulador dispõe de uma gama para V_{in} definida entre 1,2 V e os 37 V e permite uma corrente máxima de 1,5 A, com 1% de regulação da tensão de saída. Várias topologias podem ser implementadas em torno deste regulador.

No caso de polarização do LED em corrente, utiliza-se o LM317 numa configuração de fonte de corrente. A Figura 3.14 ilustra essa topologia.

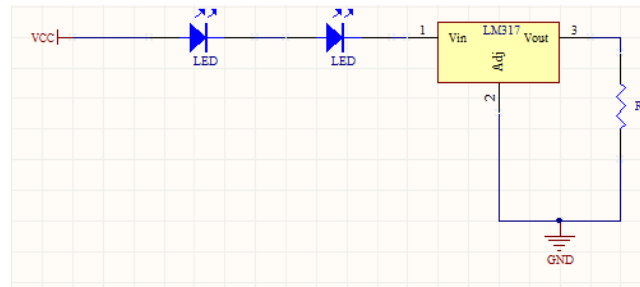


Figura 3.14 - Circuito de uma fonte de corrente usando um regulador de tensão (LM317) [4].

Segundo as especificações do dispositivo, deverá existir uma tensão entre o pino “ V_{out} ” e o pino “Adj” de 1,25 V de modo a activar a regular da tensão de saída, [24]. Com base nesta informação e atendendo à montagem, facilmente se chega à expressão que permite calcular a corrente desejada. A expressão que permite calcular a corrente resultante desta montagem é dada por,

$$I = \frac{1.25}{R} [A] \quad (3.8)$$

Considerando uma cadeia de LEDs polarizada com uma corrente de 350mA, obtém-se o seguinte valor de resistência;

$$R = \frac{1.25}{I} = \frac{1.25}{0.350} = 3,57\Omega \quad (3.9)$$

Conclui-se que esta montagem é menos sensível à variação do V_f do LED, dado que a regulação é feita somente com base no valor da resistência. Logo, possíveis variações no valor da corrente serão somente provocadas pela tolerância da resistência como demonstrado na Tabela 3.3.

V _{in} [V]	V _f [V]	R [Ω]	R-5% [Ω]	R+5% [Ω]	I (R-5%) [mA]	I (R+5%)[mA]	Potência Resistência (R-5%) [W]	Potência Resistência (R+5%) [W]	Potência LED (R-5%) [W]	Potência LED (R+5%) [W]
12	2,80	3,57	3,39	3,75	368,42	333,33	0,46	0,42	1,03	0,93
12	2,90	3,57	3,39	3,75	368,42	333,33	0,46	0,42	1,07	0,97
12	3,00	3,57	3,39	3,75	368,42	333,33	0,46	0,42	1,11	1,00
12	3,10	3,57	3,39	3,75	368,42	333,33	0,46	0,42	1,14	1,03
12	3,20	3,57	3,39	3,75	368,42	333,33	0,46	0,42	1,18	1,07

Tabela 3. 3 – Valores para o circuito com o LED Oslon SSL 150 e com o LM317 tendo em conta a tolerância da resistência.

Podemos observar na tabela que a corrente não varia com o valor de V_f , variando apenas com a tolerância da resistência. Todavia, no caso de uma cadeia de LEDs não existe variação significativa do fluxo luminoso, dado que a corrente é comum a todos eles. Do ponto de vista da potência observa-se que a dissipação na resistência é de aproximadamente 0,5 W, resultando num rendimento baixo. Uma forma de aumentar o rendimento é utilizar vários LEDs numa só cadeia. Uma fonte adicional de redução do rendimento é a potência dissipada no LM317, que terá de ser devidamente contabilizada.

Fonte de corrente com o driver da família BCR

No domínio da iluminação com recurso a LED existe, por vezes, a necessidade de se obter um circuito com topologias simplificadas. Para isso, existem vários fabricantes que disponibilizam circuitos integrados com funcionamento básico de uma fonte de corrente, mas que apresentam níveis de eficiência superior. Um exemplo de um circuito integrado para iluminação LED de baixa corrente é o 402R da Infineon, [25]. Este circuito integrado apresenta as seguintes vantagens:

- Eficiência elevada;
- Intervalo de corrente entre 20 mA a 60 mA;
- Baixo custo;
- Necessidade de poucos componentes externos;
- Fácil utilização

A figura 3.15 ilustra a topologia mais comum formada em torno do 402R.

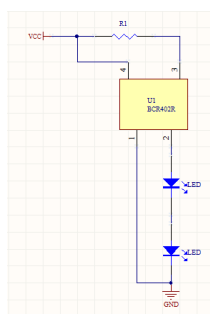


Figura 3.15 - Circuito comum utilizando o BCR402R [5].

A corrente de polarização da cadeia de LED é controlada através de uma resistência interna. Se necessitarmos de outro valor de corrente temos a possibilidade de ligar uma resistência externa em paralelo com a resistência interna, ajustando-se assim a corrente desejada. O valor da resistência externa é calculado com base no seguinte procedimento:

- Definir o valor da corrente;
- Obter o valor de V_{drop} do gráfico (mostrado em imagem a seguir) fornecido pelo fabricante na ficha de especificação, [5]

Output current versus reference voltage

$$I_{out} = f(V_{DROP}); V_S = 10V; V_{out} = 7.6V$$

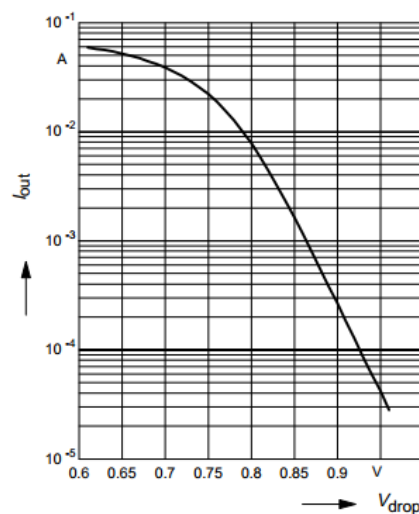


Figura 3.16 - Gráfico da corrente de saída vs tensão de referência, [5].

- Utilizar a fórmula mostrada abaixo, sendo $R_{int} = 38\Omega$, e obtermos o valor da resistência externa.

$$R = \frac{V_{drop}}{I_{out} - \frac{V_{drop}}{R_{int}}} \quad (3.10)$$

Dado que a tensão máxima de alimentação do BCR402R é de 18V, verifica-se que existe uma limitação efectiva do número de LEDs da cadeia. Todavia, a topologia inicial poderá ser modificada de modo a contornar o problema anterior, conforme se ilustra na figura 3.17.

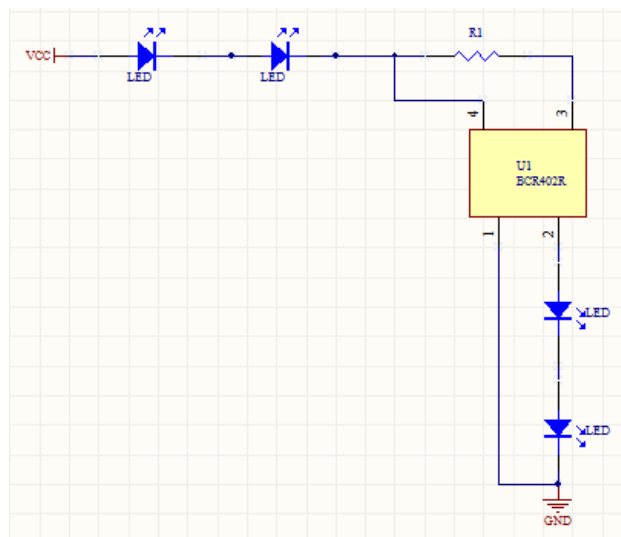


Figura 3.17 - Cálculo da resistência externa do BCR402R.

Com base nesta implementação, é possível aumentar o valor de V_{cc} visto que, se garantirmos que na entrada do BCR402R não se excede a tensão de 18V, todo o resto da tensão caíra nos LEDs que estão à esquerda do BCR402R (ver Figura 3.17).

O uso deste circuito integrado é vantajoso, dada a sua baixa complexidade associado a rendimento elevado. Esta, no entanto, é limitada à corrente máxima de 60 mA, o que para algumas aplicações não será suficiente.

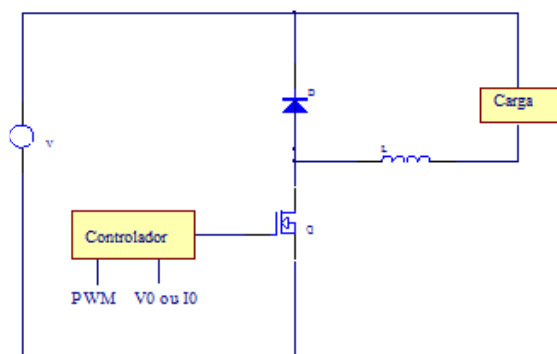
3.3. Driver com fonte comutada

Este tipo de *driver* apresenta desempenho energético superior aos anteriores, [26], podendo atingir níveis de eficiência energética entre os 75% e os 95%, [26]. As topologias mais comuns são designadas por *Buck*, *Boost* e *Flyback*.

A topologia *Buck*, ou *Step-Down*, é utilizada para os casos em que tensão de alimentação é superior à tensão da carga a controlar. Para a situação contrária à anterior é utilizada a topologia *Boost*, ou *Step-Up*. Por último, a topologia *Flyback* permite criar um isolamento galvânico entre a carga e a tensão de alimentação.

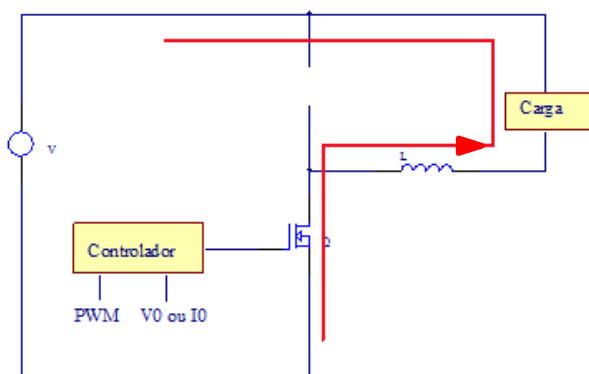
Topologia Buck (Step-Down)

O circuito mais simples que pode ser usado para implementar esta topologia é apresentado na Figura 3.18.

Figura 3.18 - Circuito típico de um driver *Buck* [26].

A constituição do circuito inclui uma bobine, um díodo FWD (*Flywell diode*), um transístor para funcionar como interruptor e um controlador.

O seu funcionamento consiste fundamentalmente em dois ciclos: o primeiro alimenta a carga e carrega a bobine, e no segundo ciclo a bobine descarrega para a carga, [26]. No primeiro ciclo, a corrente no circuito flui no sentido indicado na Figura 3.19.

Figura 3.19 - Primeiro ciclo de um driver *Buck*.

Dado que o transístor se encontra na condução, imposto pelo controlador, a corrente flui pela carga, passa pela bobine e faz com que esta carregue, [27]. No segundo ciclo, a corrente no circuito flui no sentido indicado na Figura 3.20.

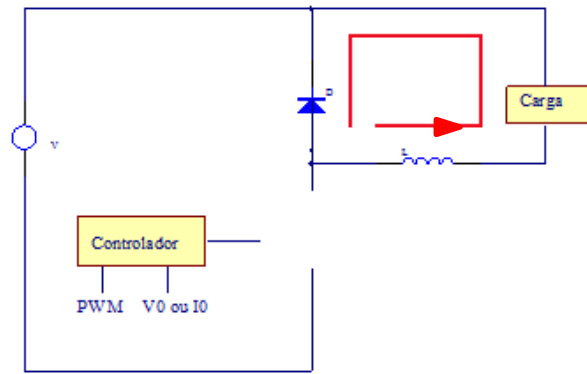


Figura 3.20 - Segundo ciclo de um driver *Buck*.

Dado que o transístor está na zona de corte, imposto pelo controlador, a corrente flui pela carga, e pelo díodo que passa a estar directamente polarizado. A energia armazenada pela bobine no ciclo anterior é devolvida ao circuito conforme é representado na Figura 3.21, [27].

Considerando que T_{on} é o tempo em que o transístor está a conduzir e T_{off} é o tempo em que o transístor não está a conduzir, o período do sinal de controlo é dado, [28],

$$T = T_{on} + T_{off} [s] \quad (3.11)$$

cujo o inverso corresponde à frequência de comutação do *driver*.

Devido ao efeito da bobine e ao sinal de controlo, a corrente no transístor tem o andamento representado na Figura 3.21.

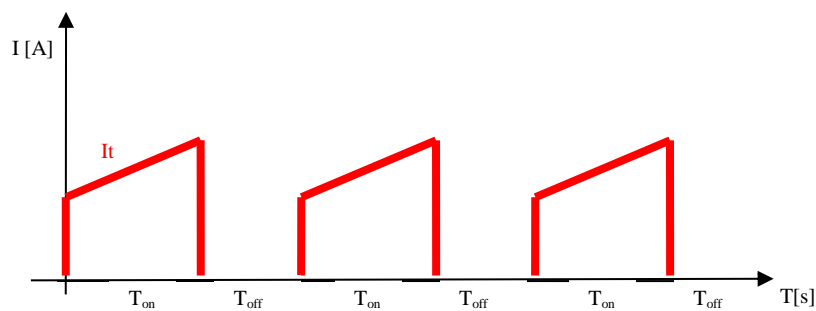


Figura 3.21 - Forma de onda da corrente no transístor durante o primeiro ciclo na topologia *buck*.

Por sua vez, o andamento da corrente no díodo está representado na Figura 3.22.

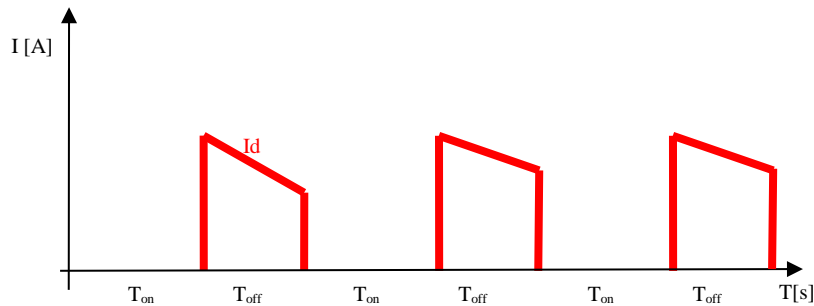


Figura 3.22 - Formas de onda da corrente no diodo durante o segundo ciclo na topologia buck.

A bobine é um dispositivo que armazena energia, ou seja, no caso do funcionamento num circuito *Buck* esta vai carregar no primeiro ciclo e vai descarregar no segundo ciclo.

No primeiro ciclo a bobine vai carregar, e a evolução da sua corrente segue

$$I_l = \frac{1}{L} \int V_l dt \quad (3.12)$$

onde

$$V_l = V - V_0 \quad (3.13)$$

o que corresponde à diferença da tensão de alimentação e a tensão da carga.

Quando o primeiro ciclo acaba, temos energia armazenada na bobine. Na passagem para o segundo ciclo, ou seja quando o transístor passa a não conduzir, a bobine vai devolver parte da energia que tem armazenada. Essa energia em forma de corrente, vai passar pelo diodo, que ficou directamente polarizado, e pela carga. Ou seja, a corrente na bobine vai começar a decrescer. O resultado dos ciclos carga e descarga da bobine em termos de corrente e tensão estão representadas na Figura 3.23.

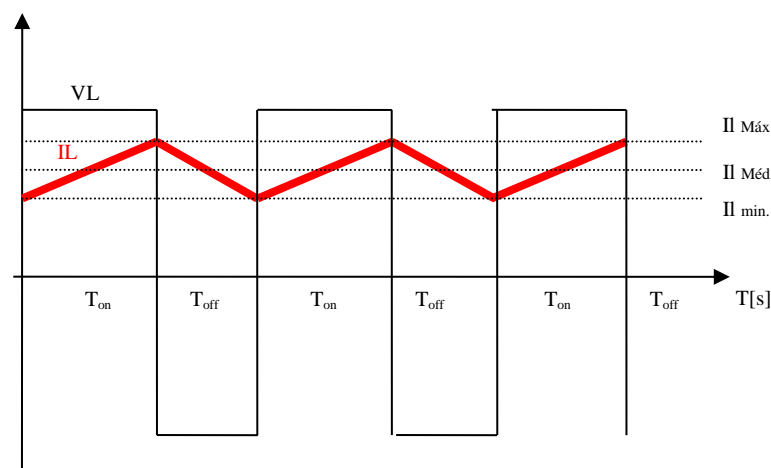


Figura 3.23 - Formas de onda na bobine.

A corrente que atravessa a bobine, vai ser a mesma que atravessa a carga, [28]. Vamos ter uma forma de onda que apresenta um *ripple*, cujo valor médio é dado por

$$I_{méd} = \frac{1}{T} \int_0^T i(t) dt \quad (3.14)$$

Por exemplo, para uma corrente de 200 mA com um *ripple* de 10%, I_{max} vai ser 220 mA e o I_{min} vai ser de 180 mA. O valor da tensão que temos presente na bobine será de,

$$V_0 = -V_l \quad (3.15)$$

Outro aspecto a considerar é o *duty cycle*, que determina a relação entre T_{on} e o T_{off} .

Dado que a tensão da bobine é

$$V_l = L \frac{di_l}{dt} \quad (3.16)$$

e,

$$\frac{di_l}{dt} = \frac{V - V_0}{L} \quad (3.17)$$

$$di_l = \frac{V - V_0}{L} dt \quad (3.18)$$

Obtém-se a variação da corrente quando o transístor está a conduzir sendo,

$$di_{l \text{ transistor}=on} = \frac{V - V_0}{L} T_{on} \quad (3.19)$$

Na situação do transístor ao corte Obtém-se,

$$\frac{di_l}{dt} = \frac{-V_0}{L} \quad (3.20)$$

$$di_l = \frac{-V_0}{L} dt \quad (3.21)$$

correspondendo à variação da corrente quando o transístor não está a conduzir dada por

$$di_{l \text{ transistor}=on} = \frac{-V_0}{L} (1 - D)T \quad (3.22)$$

onde T é o período total dos dois ciclos e D é o duty cycle.

Podemos então dizer que toda a corrente armazenada na bobine é igual a toda a corrente fornecida no ciclo em que o transístor está ao corte se estivermos em regime contínuo. Então podemos dizer que:

$$di_{l \text{ transistor}=on} + di_{l \text{ transistor}=off} = 0 \quad (3.23)$$

Substituindo,

$$\frac{V - V_0}{L} T_{on} - \left(\frac{-V_0}{L}\right) (1 - D)T = 0 \quad (3.24)$$

Obtemos,

$$V_0 = DV \quad (3.25)$$

Então a expressão do *duty cycle*, que representa também a sua função de transferência, fica:

$$D = \frac{V_0}{V} \quad (3.26)$$

Onde V_0 é a tensão da carga, e V é a tensão de alimentação.

Outro pormenor que devemos ter em conta é o modo de funcionamento, podemos ter um funcionamento contínuo, descontínuo ou no limite da continuidade. Tradicionalmente chama-se ao modo contínuo CCM (*Continuous Conduction Mode*), ao modo descontínuo DCM (*Discontinuous Conduction Mode*) e ao modo no limite da continuidade BCM (*Boundary Conduction Mode*). A principal diferença entre estes modos será até onde chega a corrente na fase de descarga da bobine. As Figuras 3.24, 3.25 e 3.26 mostram as diferenças entre os modos.

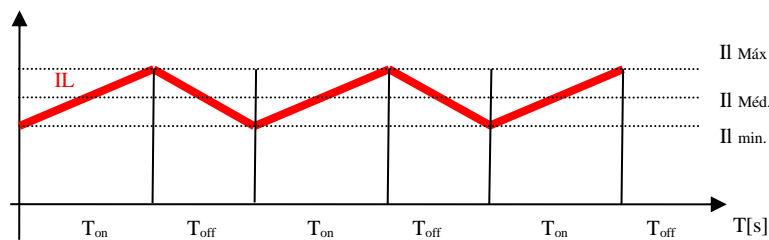


Figura 3.24 - Formas de onda na bobine no modo CCM.

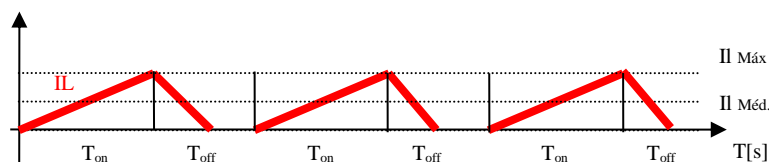


Figura 3.25 - Formas de onda na bobine no modo DCM.

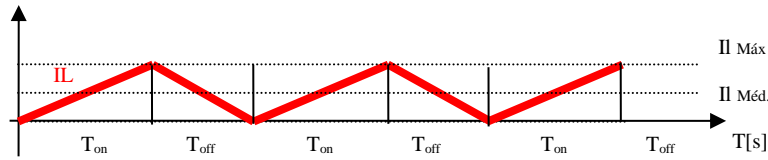


Figura 3.26 - Formas de onda na bobine em modo BCM.

No modo CCM a corrente na bobine nunca atinge o valor zero. Este tipo de modo é o mais usado para drivers de LED, dado que permite obter valores médios de corrente mais altos e atingir eficiências mais elevadas, [28].

No modo DCM a corrente atinge o valor zero durante um pequeno período dentro do tempo que o transistor está ao corte, permitindo reduzir o valor médio da corrente. Pode também ser usado quando existem requisitos de corrente nula no início do ciclo de carga da bobine, [28].

No modo BCM, estamos no limiar entre o modo CCM e o modo DCM. Ou seja, neste modo a corrente atinge zero no fim do tempo que o transistor está ao corte e começa a carregar a partir de zero quando o transistor começa a conduzir. Este modo não é muito usado, a não ser que seja necessário, isto porque uma pequena variação pode fazer com que o driver funcione em CCM ou em DCM, [28]. Isto não é aconselhável porque para cada modo existem diferentes algoritmos de controlo, [28].

Contudo, este modo auxilia a calcular o valor da bobine que poderá determinada através de,

$$L_b = \frac{(1 - D)R}{2f} \quad (3.27)$$

onde D é o *duty cycle*, R é a resistência de carga e f é a frequência de comutação do driver,

$$f = \frac{1}{T} \quad (3.28)$$

Para funcionar em modo CCM é necessário que $L_b < L_{usada}$, e por sua vez para funcionar em DCM $L_b > L_{usada}$.

O controlo do débito de corrente e tensão do driver pode ser feito de duas formas, através da tensão de saída ou da corrente. O que foi dito atrás aplica-se em ambas, sendo que em vez de termos V_0 , teremos de obter uma tensão a partir da corrente de saída. Isto pode ser feito, como é comum nos drivers LED, com a utilização de uma resistência R_{sense} . A Figura 3.27 mostra uma forma de se implementar ambas as situações.

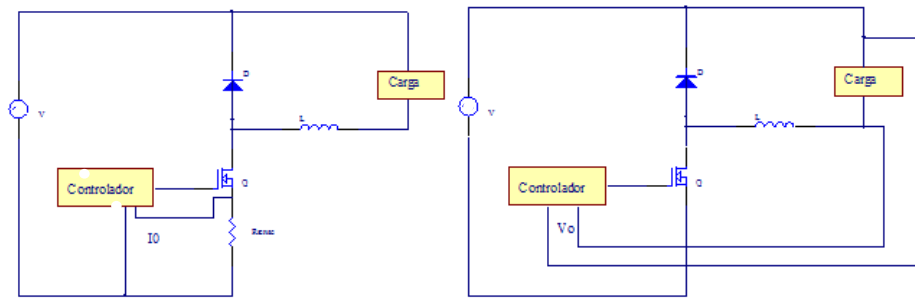


Figura 3.27 - Implementação da forma de controlo por tensão, à direita, e forma de controlo por corrente, à esquerda.

Para além do controlo ser feito por tensão ou por corrente, uma forma de controlo auxiliar é através de PWM. Este controlo permite fazer o controlo de brilho dos LEDs. O sinal PWM é constituído por uma onda rectangular entre $V+$ e zero, com um determinado duty cycle. A Figura 3.28 ilustra várias formas de onda PWM com diferentes duty cycles.

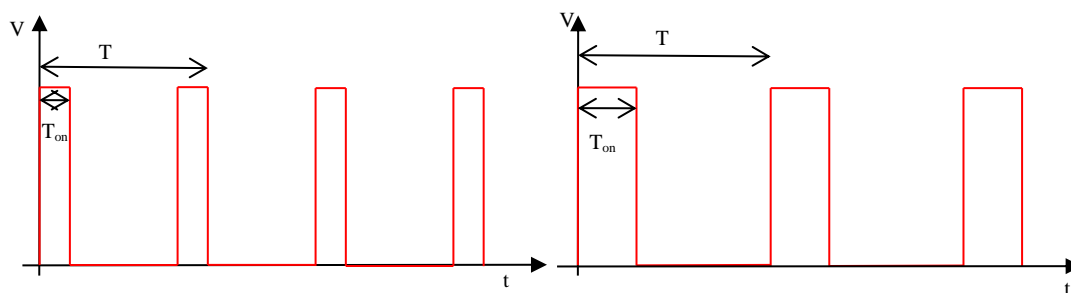


Figura 3.28 - Formas de onda PWM com diferentes *duty cycles*.

Na Figura 3.28 temos duas formas de onda de PWM com o mesmo período, com a diferença do tempo que estão a *on*, ou seja, o duty cycle dado por

$$D = \frac{T_{on}}{T} \quad (3.29)$$

Ao controlar o duty cycle do driver, ajusta-se a variável responsável pelo controlo da tensão ou da corrente na carga. A Figura 3.29 mostra um caso de aplicação deste tipo de controlo.

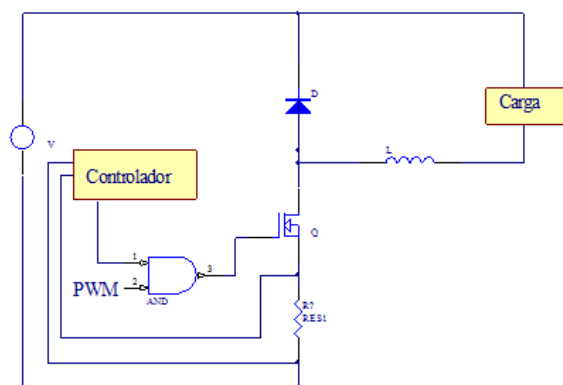
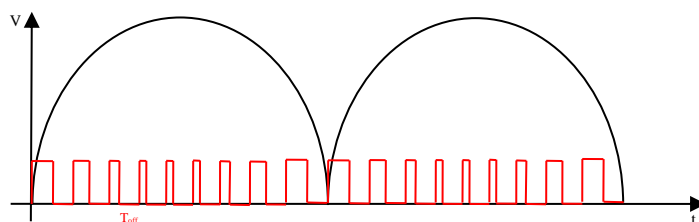


Figura 3.29 - Implementação de controlo por PWM.

Nas aplicações onde a fonte de alimentação é AC adopta-se a técnica da frequência de comutação variável, com T_{off} constante. Tem como base de funcionamento manter o T_{off} constante, e consoante a tensão de entrada varia o T_{on} também varia. Por exemplo na rede eléctrica em Portugal, a tensão eficaz é de 230 V com uma frequência de 50 Hz. Se quisermos aplicar uma tensão deste género num driver *Buck*, o ideal será aplicar essa tensão com uma rectificação de onda completa, passando a frequência para 100 Hz. Na figura seguinte mostra-se o resultado, com escala não real devido à frequência alta de comutação necessária, de um circuito com frequência variável e tempo de off constante:

Figura 3.30 - Representação do funcionamento em frequência variável e t_{off} constante.

Na Figura 3.30, a vermelho está representada a onda de controlo do transístor, que denota que a frequência está a variar consoante o andamento da tensão de alimentação que está representada a preto. Como o tempo a off é constante, o factor que varia é o tempo a *on*. Isto porque se, a variação da corrente na bobine, depende da tensão aos terminais da mesma.

Driver Buck usando o circuito integrado LT3590

Para demonstrar o resultado de um driver LED usando a topologia de um Buck, e recorrendo ao uso de um *software* da Linear Technology, o LTSpice, apresenta-se um conjunto de resultados de simulação obtidos com circuito integrado LT3590 da Linear Technology, que tem como função fazer o controlo de uma série de LEDs, [29]. É um driver que tem uma tensão de entrada que pode ir dos 5.5 V até aos 55 V, tem o transístor de comutação interno com uma corrente máxima de 80 mA, [29]. Este circuito integrado tem ainda o díodo FWD integrado. Este circuito tem uma frequência de operação fixa, de

850 kHz, [29]. Este integrado acaba por ser uma solução económica visto que contém na sua constituição alguns componentes que em outros são necessários usar como elementos externos, podendo ir até aos 91% de eficiência quando alimentado a 48 V e alimentando 10 LEDs. O circuito típico de utilização deste circuito integrado com uma tensão de alimentação de 48V, e para uma corrente de 50 mA nos 10 LEDs é apresentado na Figura 3.31.

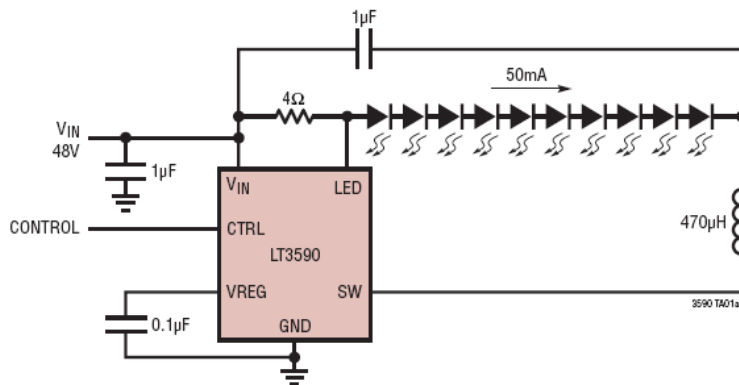


Figura 3.31 - Circuito típico de utilização do LT3590 [10].

Usando o LTspice, e fazendo a análise do comportamento do circuito Obtém-se os resultados apresentados na Figura 3.32.

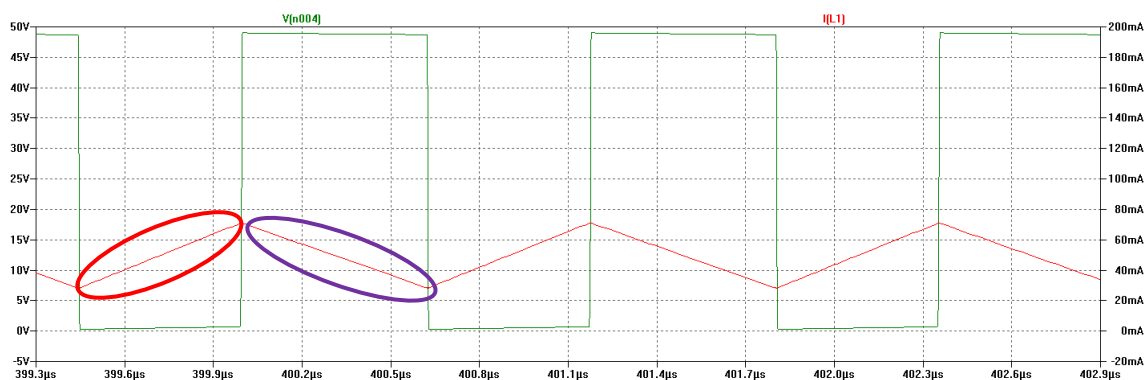


Figura 3.32 - Formas de onda da comutação e da corrente da bobine.

Como o transístor é interno e dada a construção do circuito integrado, a forma de onda de *On* e *Off* do transístor está invertida ao explicado nos pontos anteriores. Ou seja, quando a tensão no pino SW chega a zero (onda a verde), ou seja o transístor está conduzindo, logo a corrente na bobine (onda vermelha) começa a subir (elipse vermelha), ou seja, começa a carregar. Quando a tensão no pino SW chega ao máximo de tensão, ou seja o transístor está ao corte, a corrente da bobina começa a decrescer (elipse roxa) ou seja a bobine começa a descarregar. Os tempos que estão aqui em causa, tendo em conta que temos uma frequência de 850 kHz e um período de aproximadamente 1,18 µs, temos um T_{on} aproximadamente de 549 ns e um T_{off} aproximadamente de 631 ns. Isto significa que o duty cycle é de 46,5%.

Outro pormenor bastante importante de salientar é o modo de funcionamento do circuito. Podemos ver pela onda da corrente que o circuito está em CCM como modo de funcionamento, isto porque a

corrente nunca chega a zero no ciclo de descarga. Analisando a corrente vemos que o *ripple* da corrente de 20 mA o que representa 40% de *ripple*. A tensão aos terminais dos LEDs é de 21,47 V, e a corrente é de 50 mA. A potência consumida nos LEDs é de aproximadamente de 1,1 W e apresenta uma eficiência de 91% o que indica que a potência fornecida pela fonte é de 1,21 W.

Isto revela que este tipo de circuitos é bastante eficiente comparativamente a outras opções que foram apresentadas em pontos anteriores. Conseguimos passar de eficiências de 40% para os 90%.

Boost (Step-Up)

O *driver boost* tem a característica de aumentar a tensão de carga a partir da tensão de alimentação. Esta característica é o que faz com que esta topologia seja escolhida por exemplo em drivers LED que são usados em automóveis ou em aparelhos portáteis.

O circuito típico mais simples para implementação de um driver boost é apresentado na Figura 3.33, e opera em duas fases.

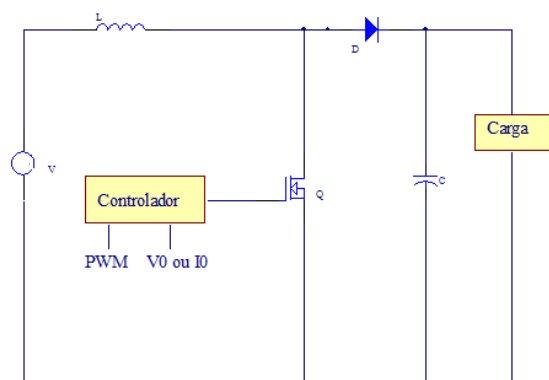


Figura 3.33 - Circuito típico de um *driver Boost*.

No primeiro ciclo a corrente flui conforme mostra a Figura 3.34.

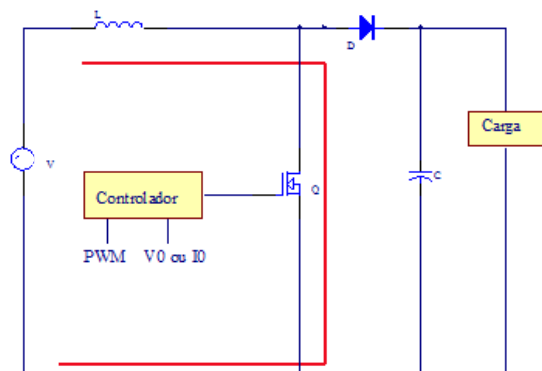


Figura 3.34 - Formas de onda no transistor durante o primeiro ciclo na topologia *boost*.

As formas de onda presentes no circuito durante o primeiro ciclo estão representadas na Figura 3.35.

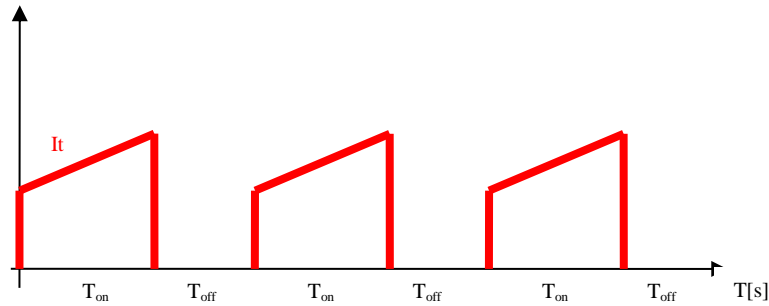


Figura 3.35 - Formas de onda da corrente no transístor durante o primeiro ciclo na topologia *boost*.

Dado que transístor está a conduzir, a corrente na bobine evolui de modo a garantir o armazenamento de energia. O valor desta corrente que atravessa o circuito no primeiro ciclo é dado por ,

$$di_{l\ 1^{\circ}\ ciclo} = \frac{V}{L} T_{on} \quad (3.30)$$

No segundo ciclo a corrente flui conforme mostra a Figura 3.36.

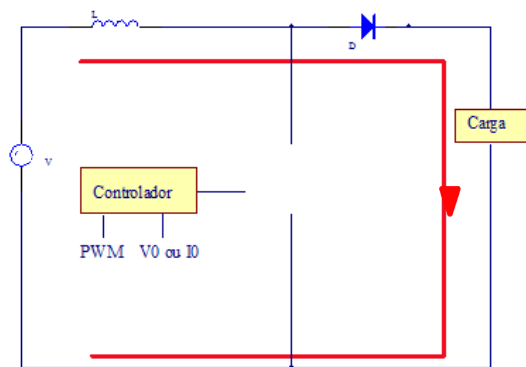


Figura 3.36 - Circulação da corrente no segundo ciclo de um *driver boost*.

Como o transístor não está a conduzir, temos a corrente a passar pela bobine, pelo diodo e pela carga. Neste ciclo a bobine actual como uma segunda fonte complementar. O valor da corrente no segundo ciclo é dado por,

$$di_{l\ 2^{\circ}\ ciclo} = \frac{(V - V_0)T_{off}}{L} \quad (3.31)$$

As formas de onda presentes no circuito durante o primeiro ciclo estão representadas na Figura 3.37.

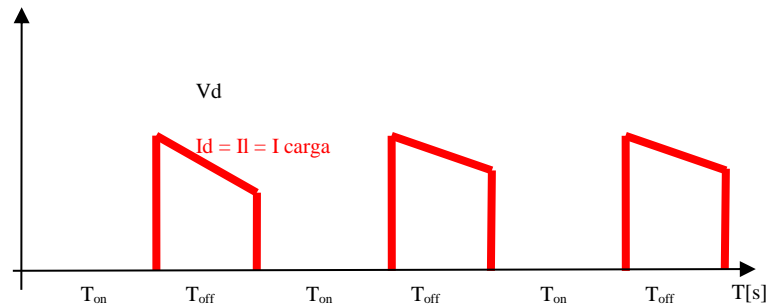


Figura 3.37 - Formas de onda no transístor durante o segundo ciclo na topologia *boost*.

A forma de onda resultante dos dois ciclos na bobine está representada na Figura 3.38.

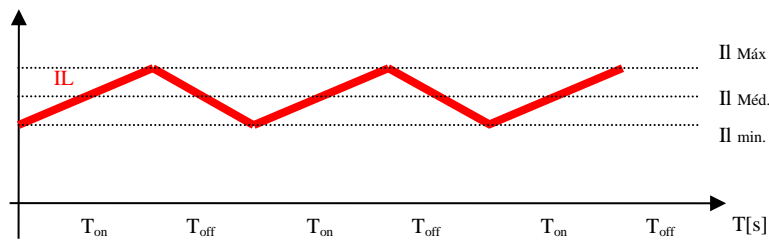


Figura 3.38 - Formas de onda da corrente resultante na topologia *boost*.

A forma de onda de tensão de saída deste *driver* está representada na Figura 3.39.

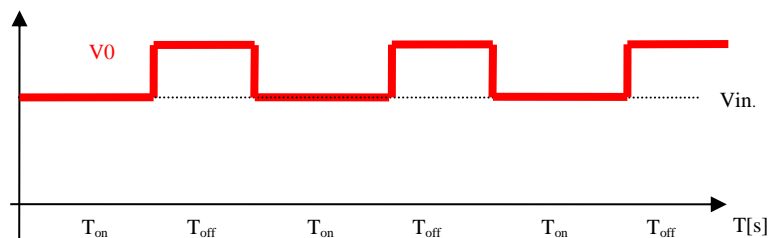


Figura 3.39 - Formas de onda da corrente resultante na topologia *boost*.

Como foi referido inicialmente, este *driver* tem uma tensão de saída superior à alimentação de entrada. Um pormenor importante na topologia *Boost*, é o facto de ser estritamente necessário a existência de um condensador à saída de modo a garantir uma saída com *ripple* reduzido, [27].

A tensão de saída relaciona-se com o *duty cycle* através de,

$$V_o = \frac{V}{1 - D} \quad (3.32)$$

onde D é o *duty cycle* associado à operação do *driver*. O *duty cycle* num *driver* do tipo *Boost* não pode ultrapassar os 50%, visto que a bobine necessita de algum tempo para descarregar a sua energia para o condensador de saída, [27]. A função de transferência do *driver boost* é dada por,

$$\frac{V_o}{V} = \frac{1}{1 - D} \quad (3.33)$$

onde D corresponde ao *duty cycle*.

Flyback

Como foi referido esta topologia é isolada. Isto significa que é incluído um isolamento galvânico entre a entrada e saída através da utilização de um transformador. Esta topologia é bastante usada na iluminação a LED, visto que permite funcionar de três formas, designadamente modo de *Step-Down*, em *Step-Up* e em *Buck-Boost*. Uma vantagem adicional desta topologia é o facto de poder disponibilizar diversas apenas com uma entrada, [27]. Estas saídas podem ser positivas ou negativas, consoante os enrolamentos do transformador, [27].

O circuito típico mais simples para implementação de um *driver flyback* é apresentado na Figura 3.40.

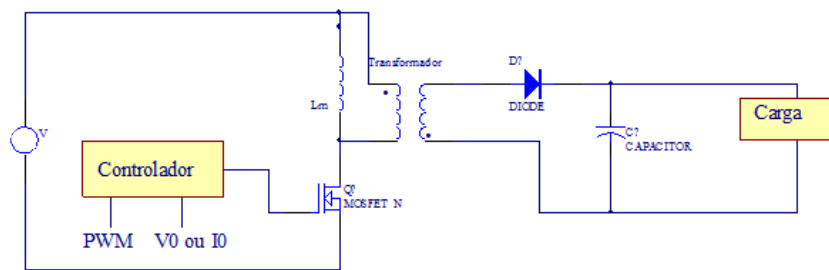


Figura 3.40 - Circuito típico de um driver *Flyback*.

O seu funcionamento é semelhante aos restantes *drivers* referidos nos pontos anteriores, [27]. Durante o primeiro ciclo o transístor está a conduzir e a corrente flui conforme mostra a Figura 3.41.

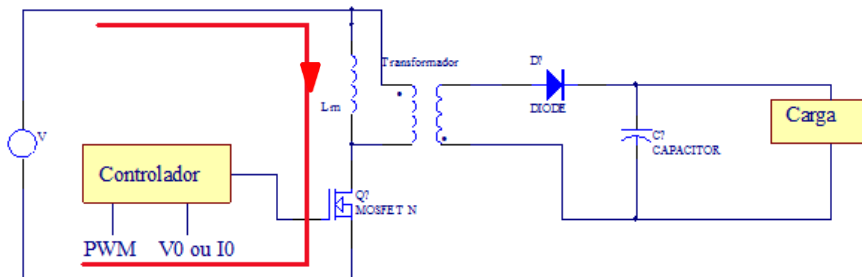


Figura 3.41 – Andamento da corrente durante o primeiro ciclo na topologia *Flyback*.

Com a corrente a fluir no primário do transformador, este está a armazenar energia. As equações seguintes mostram o valor da tensão e o valor da corrente no primário durante o primeiro ciclo,

$$V_p = L_m \frac{di_{Lm}}{dt} \quad (3.34)$$

$$i_p = \frac{V_p}{L_m} t_{on} \quad (3.35)$$

No secundário não temos nenhuma corrente a passar visto que o transformador do *flyback* está em oposição de fase, o que faz com que o díodo esteja inversamente polarizado, [27].

Durante o segundo ciclo o transístor não está a conduzir e a corrente flui conforme mostra a Figura 3.42.

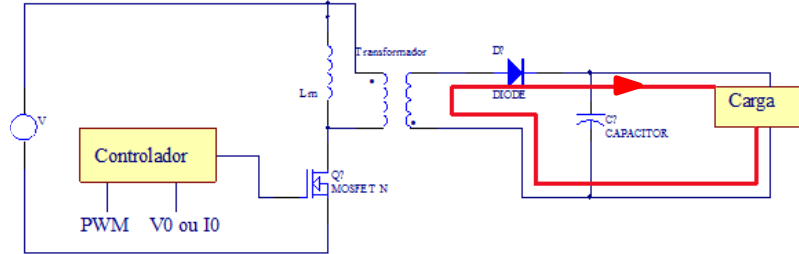


Figura 3.42 – Andamento da corrente durante o segundo ciclo na topologia *Flyback*.

Com a corrente agora a fluir no secundário do transformador, temos as equações que determinam o valor da tensão e o valor da corrente no secundário durante o segundo ciclo dadas por,

$$V_s = -V_o \frac{N_1}{N_2} \quad (3.36)$$

$$i_s = -\frac{V_o T_{off} T}{L_m} \frac{N_1}{N_2} \quad (3.37)$$

Este driver pode ter um funcionamento em modo contínuo ou em modo descontínuo, [27], dependendo do $L_{magnetizante}$ desenhado. O cálculo de $L_{magnetizante \min}$ é dado pela seguinte expressão,

$$L_{magnetizante \min} = \frac{n^2 (1 - D) Z_{carga}}{2 f_{comutação}} \quad (3.38)$$

Se $L_{magnetizante \min}$ for maior, o driver está funcionar em CCM. Conteúdo, se for menor fica a funcionar em DCM [27].

Assumindo não existirem perdas, toda a energia fornecida no primeiro ciclo é usada no segundo ciclo, [7]. Neste caso é possível afirmar que,

$$\Delta i_{Ton} + \Delta i_{Toff} = 0 \quad (3.39)$$

que se traduz em

$$\frac{V_p}{L_m} t_{on} + \left(-\frac{V_o T_{off} T}{L_m} \frac{N_1}{N_2} \right) = 0 \quad (3.40)$$

A expressão de V_o passa a ser dada por,

$$V_o = V_{entrada} \frac{D}{(1 - D)} \frac{N_1}{N_2} \quad (3.41)$$

onde D é o *duty cycle*.

Perdas em fontes comutadas

Um dos principais objetivos do dimensionamento de uma fonte comutada é que esta seja o mais eficiente possível. Para atingir uma eficiência elevada é necessário contabilizar as perdas que surgem nos componentes usados na implementação da fonte. As maiores perdas nas fontes comutadas ocorrem nas resistências, transístores e díodos, [30].

Nas resistências existem perdas por efeito de Joule, sendo que convirá reduzir ao mínimo o uso de resistências no circuito.

Existem dois tipos de perdas no transístor, nomeadamente perdas por condução e perdas por comutação. Dado que o transístor é o componente que faz a comutação entre os dois ciclos das fontes comutadas temos de ter em conta que devido à comutação em alta frequência essas perdas vão ser significativas, [30]. Na Figura 3.43 observa-se esses dois tipos de perdas num transístor MOS-FET,

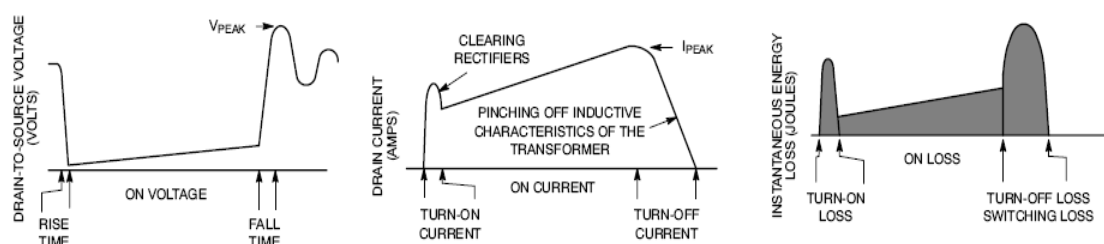


Figura 3.43 - Perdas de condução e comutação de um MOS-FET, [11].

Observando a Figura 3.43, observa-se que as perdas de comutação têm um peso significativo. Isto porque quando um transístor comuta para a condução, ocorre um pico de corrente e devido ao tempo de subida do transístor temos um tempo em que a tensão entre o dreno e a fonte do transístor desce, [30]. Ao mesmo tempo que a tensão sobe temos um pico de corrente, logo temos uma dissipação de potência. O mesmo acontece na transição para *off*, temos um tempo de descida, e durante esse instante temos um pico de tensão e um pico de corrente que leva também a uma considerável dissipação de potência, [30]. Durante a fase de condução do transístor existem perdas, porque temos, no caso de um MOS-FET, temos uma resistência de canal ($R_{ds(on)}$) sendo percorrida por uma corrente (I_d) o que provoca dissipação de potência, [30]. Uma forma de minimizar estas perdas, tanto de comutação como de condução, é escolher um transístor que tenha uma carga de gate e uma $R_{DS(on)}$ o mais baixo possível. Outra forma é não utilizar frequências de comutação muito elevadas para minimizar as perdas de comutação, [30].

Outro componente que provoca alguma ineficiência da fonte comutada é o díodo FWD. Um díodo também apresenta perdas de comutação e de condução, [30]. A Figura 3.44 ilustra estas perdas.

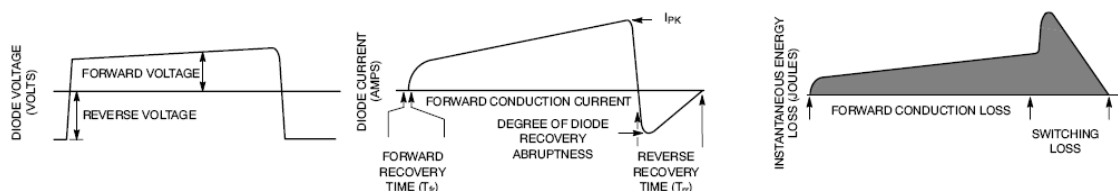


Figura 3.44 - Perdas de condução e comutação de um díodo, [11].

As perdas de condução estão relacionadas directamente com o seu V_f , o que conjugado com a corrente que o percorre origina uma dissipação de potência considerável, [30]. As perdas por comutação estão relacionadas essencialmente com a passagem do modo de condução para o modo inverso, [30]. Para minimizar estas perdas, é necessário seleccionar um díodo com um V_f de baixo valor e optar pela escolha de um díodo com recuperação rápida (*Fast Recovery*), por exemplo um *Schotky*, [30].

Factor de potência

A potência associada ao circuito electrónico reparte-se entre três tipos de potência, nomeadamente a potência aparente, potência activa, potência reactiva e a potência reactiva associada às harmónicas. A representação do triângulo das potências é ilustrada na Figura 3.45.

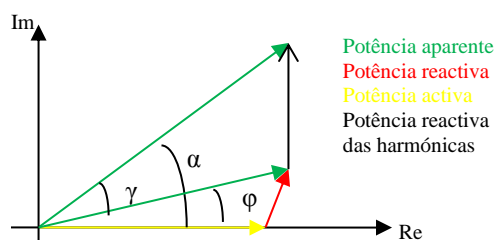


Figura 3.45 – Tetraedro de potências.

Para um circuito ter um bom desempenho será ideal que a componente de potência reactiva seja nula, ou seja, que o circuito consuma somente potência activa. Isso faz com que todos os ângulos sejam zero. O conceito de factor de potência, representa a diferença de fase entre a tensão e a corrente de entrada. Quanto mais o factor de potência se aproximar de um, mais eficaz é o circuito, [30].

Quando isso não acontece é necessário encontrar uma estratégia para melhorar o factor de potência do circuito. A esta técnica chama-se correcção do factor de potência, podendo esta ser obtida de diversas formas, tanto activas como passivas. Uma das técnicas passivas que obtém bons resultados com uso de poucos componentes é a topologia *Valley Fill*.

Esta topologia faz com que num circuito, por exemplo um *driver Buck* com um factor de potência de 0,5 fique com factores de potência que podem ir até aos 0,9 nos melhores casos, [33]. Este circuito está representado na Figura 3.46.

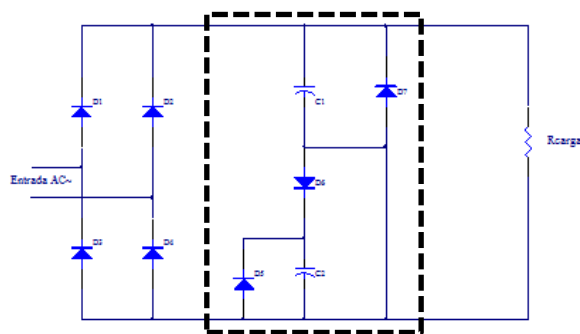
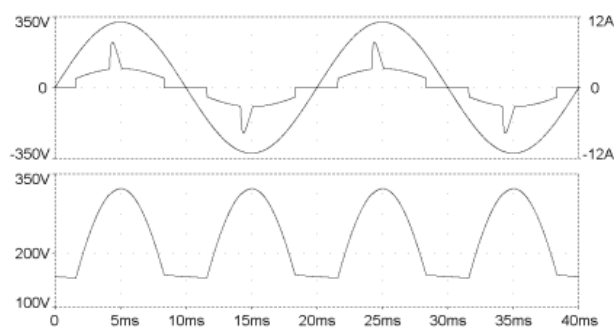


Figura 3.46 - Circuito Valley-Fill.

O funcionamento consiste em fazer com que a carga dos condensadores seja feita em série através de D4, e a descarga é feita com os condensadores em paralelo. Ou seja, como temos uma capacidade relativamente baixa durante a carga, porque temos os condensadores em série, a corrente não vai ter picos demasiados elevados, provocados por capacidades elevadas, logo não vai ter uma corrente média demasiado elevada. A corrente de entrada ideal do circuito *Valley-Fill* passa a ficar como representado na Figura 3.47.

Figura 3.47 - Formas de onda à saída do circuito *Valley-Fill*, [12].

Para minimizar o pico central de cada ciclo da onda, podemos acoplar uma resistência a seguir a D6 para evitar que o pico suba demasiado, mas isto traz uma desvantagem porque temos mais uma fonte de perdas o que pode fazer baixar a eficiência.

4. Tecnologias para Gestão de Sistemas de Iluminação

Dado o crescente enfoque na eficiência energética e tendo a iluminação um peso significativo neste âmbito, há que desenvolver estratégias de optimização do uso da iluminação. Uma luminária do tipo ON/OFF depende somente do rendimento da sua fonte de luz. No entanto, é possível reforçar o grau de optimização na eficiência de uma luminária através da integração de mecanismos de controlo e gestão inteligente. Para além destes ganhos, a incorporação de um sistema de gestão flexível e reconfigurável tem constituído um factor nuclear no desenvolvimento de edifícios inteligentes, [34, 35].

4.1. Estratégias de controlo para sistemas de iluminação

Estratégia de medição do nível de luminosidade

Uma das estratégias de controlo passa por utilizar a informação obtida através da medição do nível de luminosidade ambiente.

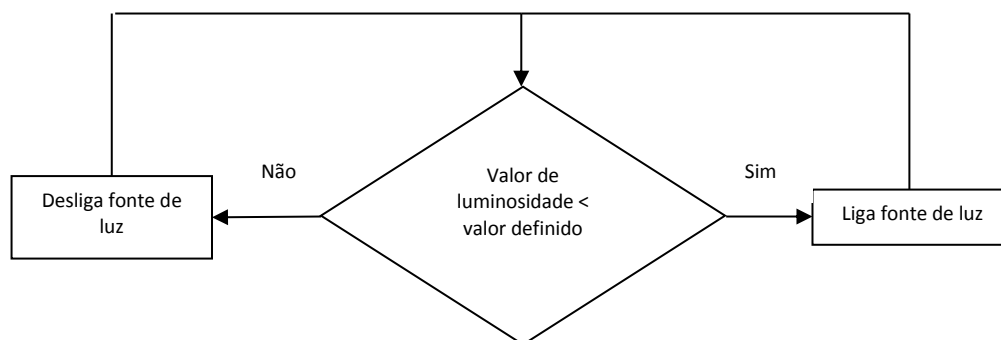


Figura 4.1 - Algoritmo de decisão dependendo do valor de luminosidade.

A Figura 4.1 apresenta um algoritmo simplificado em que a activação de uma fonte de luz é condicionada pelo nível de iluminação. [35]. A Figura 4.2 ilustra um caso de aplicação em que se recorre a este tipo de decisão.

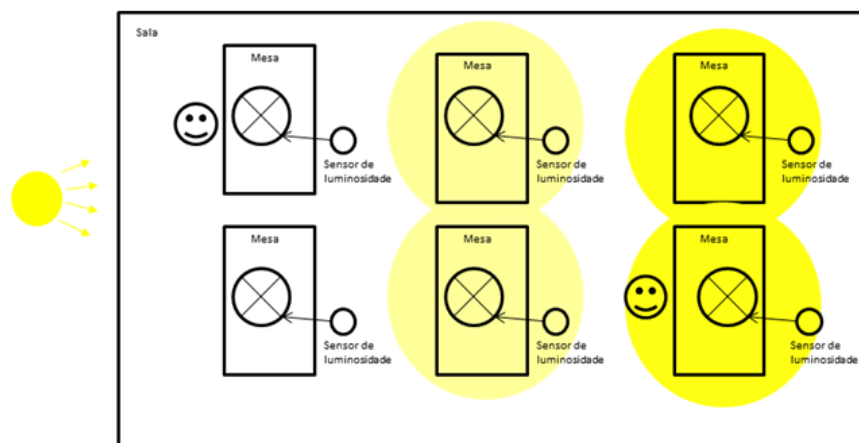


Figura 4.2 - Exemplo de aplicação do algoritmo de decisão dependendo do valor de luminosidade.

Podemos observar na figura 4.2 que a intensidade luminosa seleccionada em cada foco depende directamente da luz solar incidente (à esquerda).

O nível de conforto visual obtido desta estratégia pode ser melhorada com base na informação sobre finalidade da área a iluminar. A Tabela 4.1 apresenta os níveis de luminância definidos pela norma DIN 5035-2:1990.

Nível	Luminância	Actividade
1	15	Orientação: só estadias temporárias
2	30	
3	60	
4	120	Tarefas ligeiras com contrastes elevados
5	250	
6	500	Tarefas visuais normais com detalhes médios
7	750	
8	1000	Tarefas visuais exigentes com pequenos detalhes
9	1500	
10	2000	Tarefas visuais muito exigentes com detalhes muito pequenos
11	3000	
12	>5000	Casos especiais

Tabela 4.1 - Valores de luminância segundo a norma DIN 5035-2:1990, [9].

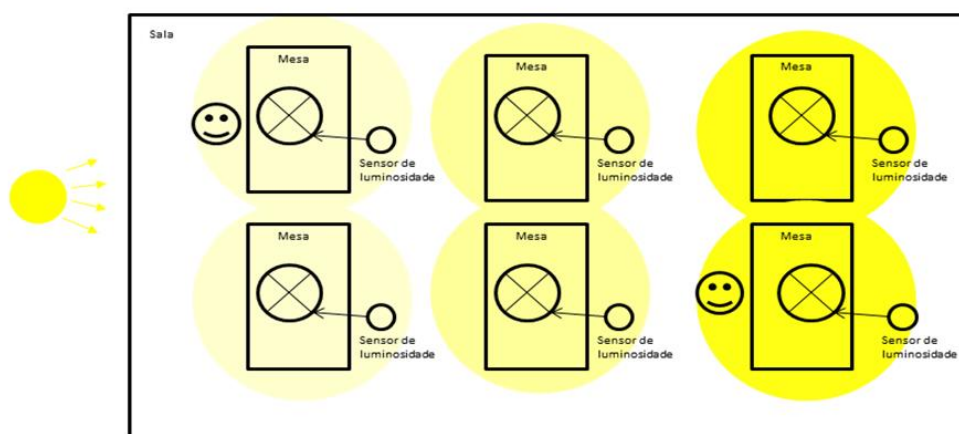


Figura 4.3 - Sistema de gestão com sensores de luminosidade locais.

Por exemplo, os níveis de iluminância num escritório (Tabela 4.1) devem garantir entre 500 e 750 LUX's na área de trabalho. Ou seja, o sistema vai fazer a leitura dos valores dos sensores ao longo do tempo, e vai gerindo a intensidade luminosa que a luminária está a emitir, [36]. Este sistema apresenta a vantagem, de equilibrar a intensidade luminosa da luminária consoante fontes de luz exteriores ao sistema, por exemplo a luz solar.

Estratégia de ocupação do espaço temporal

Um sistema de iluminação poderá incluir controlo prevendo a ocupação de um espaço, tanto por nível de ocupação real ou por estimação de ocupação e utilização . A estimativa de ocupação poderá ser tipificada através de níveis de utilização do espaço consoante um horário pré-definido. O exemplo da Figura 4.4 representa um sistema de gestão com base em programação horária.

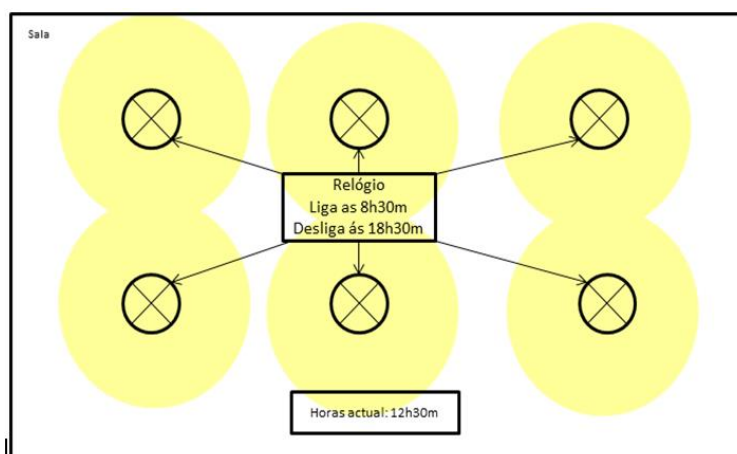


Figura 4.4 - Sistema de gestão com estimação de ocupação com base em programação horária.

O exemplo mostra que o sistema gere a iluminação por uma hora definida, mas poderemos ter sistemas que façam a gestão da iluminação com base em outro parâmetro a definir consoante a instalação pretendida.

Por vezes temos a dificuldade de garantir que um espaço tenha uma utilização com um parâmetro fixo, como por exemplo um horário. Para ultrapassar essa questão, temos a possibilidade de gerir um sistema de iluminação com base na presença efectiva no espaço. Ou seja, a iluminação vai ser gerida ao longo do espaço consoante o nível de ocupação. Assim sendo, garantimos que iluminamos uma determinada área de um determinado espaço somente quando é necessário. Esta gestão pode ser efectuada com base no uso de sensores de presença, e em algumas aplicações, prevê o uso de algoritmos que fazem com que haja a previsão de utilizações de áreas, [37]. Na Figura 4.5 ilustra-se uma aplicação de um sistema de gestão que usa a estratégia de ocupação real do espaço.

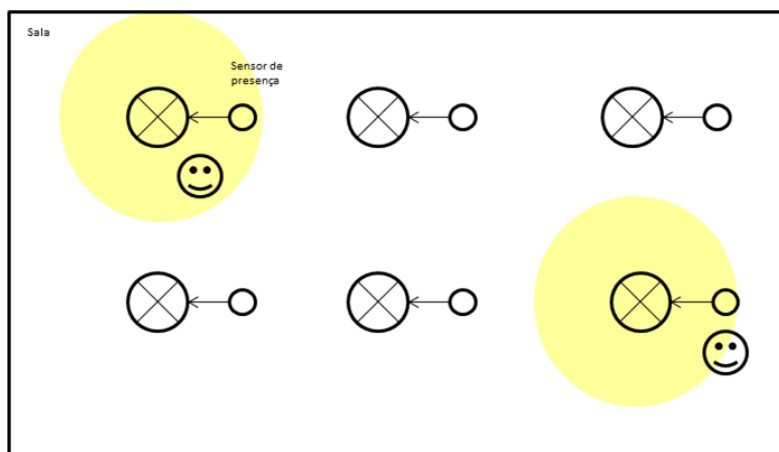


Figura 4.5 - Sistema de gestão com controlo de ocupação real com base em uso de sensores de presença.

Este sistema tem o funcionamento básico de ter pontos iluminados onde de facto existe a ocupação desse mesmo espaço, o que faz com que espaços que não tenham ocupação real não haja iluminação, o que implica que não haverá desperdício energético. Isto é uma vantagem relativamente à estratégia anterior, eliminando assim a desvantagem principal do ponto anterior. Uma possível desvantagem será o custo do sensor de presença que poderá ser dispendioso dependendo da tecnologia usada, o que pode tornar o sistema total dispendioso.

Estratégia de gestão complementar

Um sistema de gestão de iluminação tendo uma base de estratégia complementar, passará por usar a combinação de estratégias referidas nos pontos anteriores, conforme se ilustra na Figura 4.6.

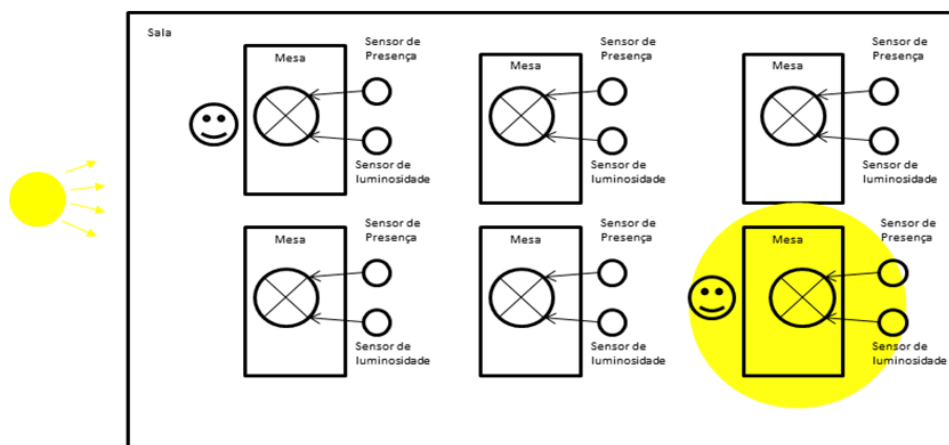


Figura 4.6 - Sistema de gestão com estratégia composta com base em uso de sensores de presença e de luminosidade.

O funcionamento consiste em fazer com que na área de trabalho exista um nível de luminosidade definido consoante a presença. Se observarmos a Figura 4.6, existem áreas não ocupadas em que as luminárias estão apagadas ou em modo de baixa luminosidade. Esta estratégia tem a vantagem de baixar consideravelmente o consumo energético, visto que otimiza o tempo de operação da luminária.

Todavia a adição de sensores e de controlo inteligente reflecte-se num acréscimo do custo inicial do sistema.

Quadro comparativo entre estratégias

Apresenta-se na Tabela 4.2 um quadro comparativo entre estratégias.

Estratégia	Nível de luminosidade	Estimação de ocupação	Ocupação real	Luminosidade constante	Composta
Vantagens	Muito baixo custo Instalação fácil Eficiência energética moderada	Custo baixo Instalação moderada Eficiência energética moderada	Eficiência energética elevada	Baixo custo	Eficiência energética muito elevada
Desvantagens	Desperdício energético em períodos desnecessários	Desperdício energético em períodos desnecessários Acerto horário necessário	Custo elevado	Eficiência energética limitada às condições desejadas Algoritmo pode ser complexo	Custo muito elevado
Locais de aplicação	Escolas Lojas Fabricas Museus	Escolas Lojas Fabricas Museus	Escadas de serviço Instalações sanitárias Armazéns Escritórios	Salas de aula Escritórios Salas de desenho	Salas de aula Escritórios

Tabela 4.2 - Comparação entre estratégias de gestão.

4.2. Arquitecturas de controlo

A implementação do sistema de gestão, atrás referido, assenta numa arquitectura que podem variar entre:

- Arquitectura de gestão autónoma,
- Arquitectura de gestão por zona,
- Arquitectura de gestão centralizada e
- Arquitectura de gestão complementar.

Gestão autónoma

A arquitectura de um sistema de gestão autónoma pode ser constituído segundo a Figura 4.7.

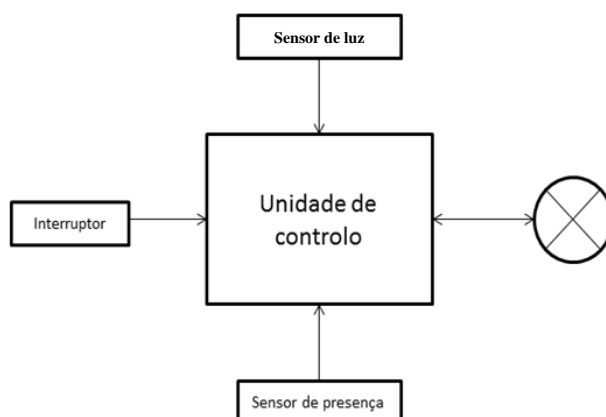


Figura 4.7 - Arquitectura de um sistema de gestão isolada.

Este tipo de arquitectura é utilizado quando se pretende que cada luminária se gira a si própria. A sua constituição pode variar consoante a estratégia a implementar, ou seja, pode incluir na sua constituição um sensor de presença, um sensor de luminosidade que permite utilizar estratégias como nível de luminosidade, ocupação real ou luminosidade constante, [35]. Na Figura 4.8 é ilustrada uma aplicação com várias luminárias que usam gestão autónoma.

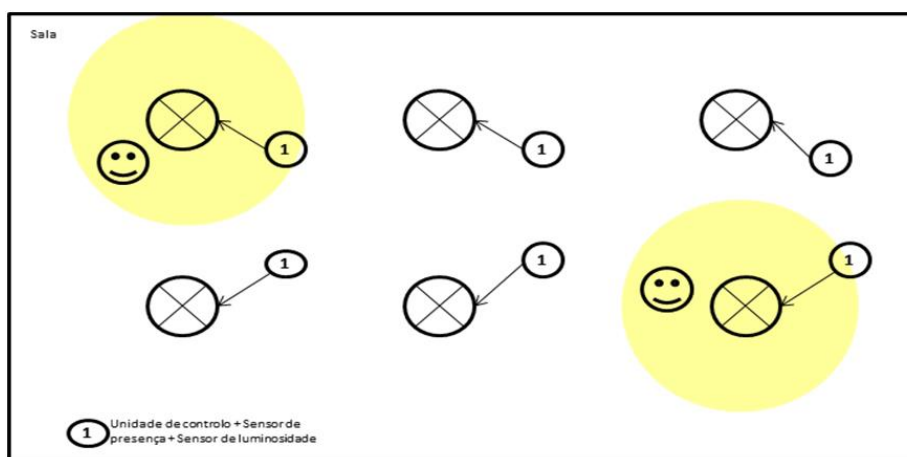


Figura 4.8 - Exemplo de uso da arquitectura de um sistema de gestão autónoma.

Esta arquitectura permite uma fácil instalação das luminárias, conferindo alguma inteligência ao sistema de iluminação. Contudo, não promove a interacção entre luminárias.

Gestão geral zona a zona

Este tipo de arquitectura é usado quando pretendemos ter um espaço em que as luminárias interagem umas com as outras permitindo assim que o espaço em questão tenha um ambiente controlado. A sua constituição pode variar consoante a estratégia a implementar, ou seja, pode ter na sua constituição um sensor de presença, um sensor de luminosidade ou um módulo de gestão central que permite gerir

zonas independentes, [37]. Esta arquitectura permite utilizar estratégias como nível de luminosidade, estimação de ocupação, ocupação real ou luminosidade constante, [35].

Nas Figuras 4.9 e 4.10 são apresentados casos de aplicação de várias luminárias com gestão zona a zona e gestão zona a zona centralizada, respectivamente.

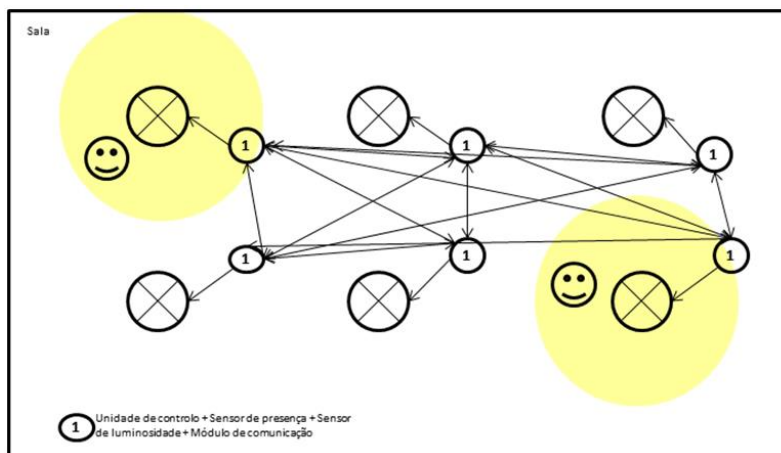


Figura 4.9 - Exemplo de uso da arquitectura de um sistema de gestão zona a zona sem gestão centralizada.

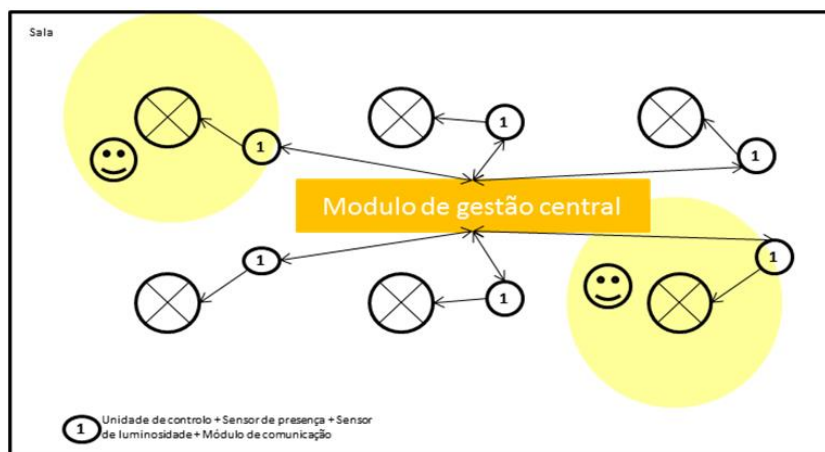


Figura 4.10 - Exemplo de uso da arquitectura de um sistema de gestão zona a zona com gestão centralizada.

Em ambas, o funcionamento consiste em cada luminária na leitura do ambiente em redor através dos sensores que têm acoplados de modo a actuar em conformidade, comunicando de seguida o seu estado. As luminárias podem também interagirem entre si complementando as acções a realizar por cada uma. Suponhamos que a luminária nº1 detecta pouca luminosidade e que por isso deve entrar em funcionamento. Ao comunicar a sua mudança de estado, as luminárias em seu redor vão responder entrando em funcionamento a 100% ou numa intensidade mais baixa para iluminarem a sua área e completarem a iluminação da área abrangida pela luminária nº1. Este é o caso prático desta tese, ou seja, as luminárias vai reagir conforme o estado das luminárias em seu redor.

A comunicação entre elas respeita um protocolo, e pode ser feita directamente entre luminárias ou pode ser coordenado através de um módulo central que gere todas as luminárias e reencaminhar a informação para as luminárias desejadas. Uma vantagem do uso desta arquitectura é permitir uma fácil instalação das luminárias dependendo do protocolo de comunicação, e confere uma elevada inteligência ao sistema de iluminação. Outra vantagem será o facto de se poder obter elevados desempenhos no que diz respeito à eficiência energética, [35].

Gestão centralizada

Este tipo de arquitectura é muito semelhante à gestão zona a zona com gestão centralizada, com a diferença que a gestão não é feita somente numa zona mas sim num sistema inteiro. Na Figura 4.11 mostra-se um caso de aplicação deste tipo de arquitectura.

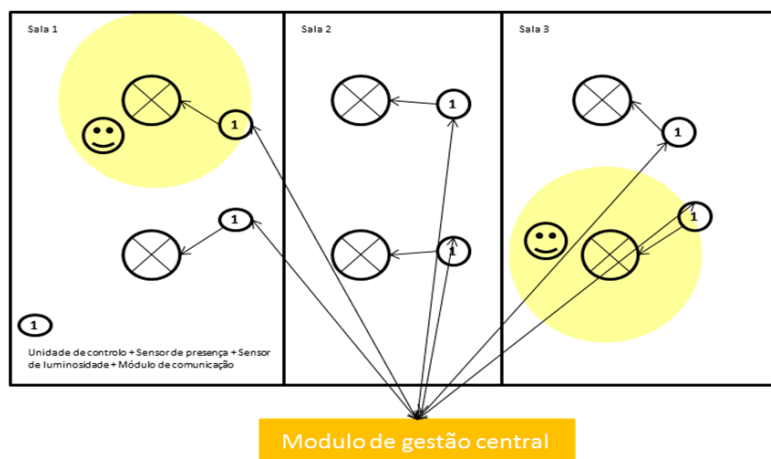


Figura 4.11 - Exemplo de uso da arquitectura de um sistema de gestão centralizada.

Tomando como exemplo sistemas de presença e localização, que podem estar ligadas a um sistema de segurança, esta arquitectura permite para além de iluminar a área, informar onde existe presença de pessoas. Outro exemplo será o facto de se prever a manutenção da instalação visto que uma falha em uma luminária num local qualquer, rapidamente será localizada pelo sistema central e assim torna-se mais rápida a acção de manutenção. A possibilidade de integração com outros sistemas e funcionalidades extra, revela uma enorme vantagem relativamente às arquitecturas mostradas anteriormente. Outra vantagem será a eficiência do sistema. Mas esta arquitectura traz também algumas desvantagens. Uma delas será a complexidade da instalação, a sobre carga de processamento do sistema central dependendo do número de áreas a gerir e complexidade do algoritmo de funções que pode provocar alguns atraso em reacções nas luminárias, [35].

Gestão complementar

Esta arquitectura é composta por duas arquitecturas referidas em pontos anteriores. Temos a junção da arquitectura de gestão zona a zona centralizada e a arquitectura centralizada. Na Figura 4.12 temos um caso de exemplo de aplicação da arquitectura composta.

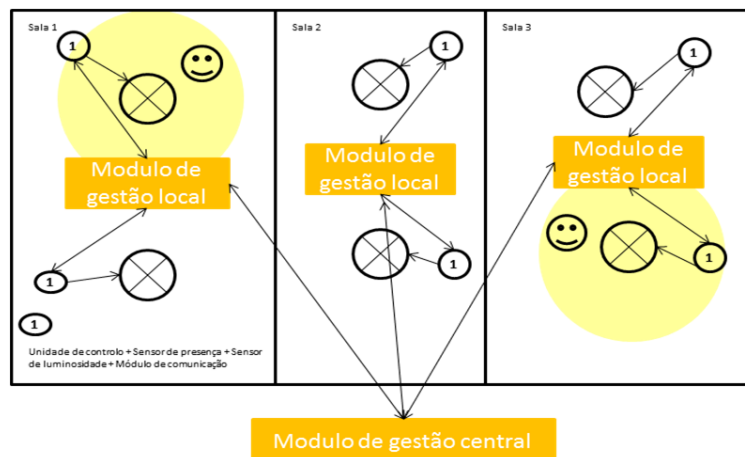


Figura 4.12 - Exemplo de uso da arquitectura de um sistema de gestão composta.

4.3. Tipos de comunicação no domínio da iluminação inteligente

Nos pontos seguintes serão enumerados os principais protocolos de comunicação usados nos actuais sistemas de iluminação.

DMX512

DMX512 é o acrónimo de *Digital Multiplex*, e é um protocolo de comunicação muito usado em controlo de iluminação de entretenimento no controle da iluminação móvel e em *dimmers*.

A primeira entidade a patentear o DM512 foi o USITT (United States Institute of Theathre Tecnology) em 1986, [38]. Em 1998 a manutenção deste protocolo ficou a cargo da ESTA (The Entertainment Services and Technology Association) garantindo assim que o protocolo continuava em desenvolvimento [38]. Em 2004 a ANSI (American National Standards Institute) aprovou o DMX 512 como *standard* ANSI E1.11-2004.

O DMX512 tem como base o EIA/TIA-485, ou seja, é um canal bidireccional ou half-duplex que tem a capacidade de ter ligados 32 equipamentos no mesmo BUS com um comprimento máximo de 1200 metros, e com uma taxa de transferência máxima de 10 Mbps, [39]. No caso específico do DMX 512 terá de existir um repetidor a cada 500 m, [40].

Num sistema DMX512 só é aconselhado a interligação entre equipamento em *Daisy-chain*. Outros tipos de ligações como estrela, *ring*, *mesh* não estão previstas, [39] A Figura 4.13 mostra como se forma um sistema DMX512.

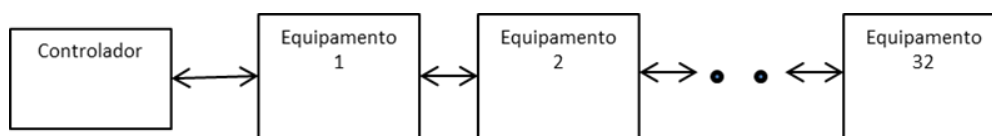


Figura 4.13 - Exemplo de interligação de um sistema com protocolo DMX512.

O sistema é interligado entre todos os equipamentos, no máximo 32, e cada equipamento faz a passagem de informação para o seguinte até se chegar ao final. Se a terminação existir, o valor tabulado será uma terminação com uma impedância de 120 Ω .

A taxa de transferência do DMX 512 é de 250 kbps e a estrutura dos pacotes enviados está representado na Figura 4.14.

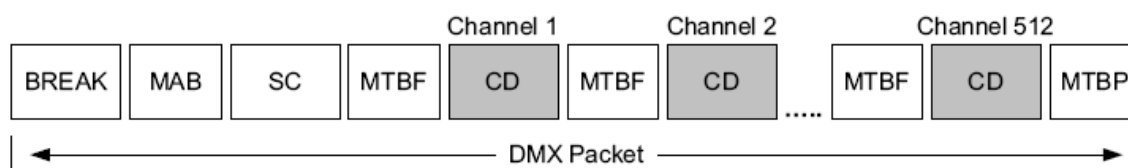


Figura 4.14 - Estrutura de um pacote DMX512.

DALI (Digital Addressable Lighting Interface (IEC 60929 Annex E))

O sistema DALI é um bus e um protocolo muito utilizado em sistemas de iluminação inteligente, [42]. Este protocolo padrão é definido pela norma IEC 60929 Anexo E, que não depende de nenhum fabricante específico. Ou seja, qualquer fabricante pode usar e desenvolver o sistema, permitindo assim a integração de vários módulos de fabricantes distintos, [43].

O sistema permite controlar luminárias individuais ou em grupo, nomeadamente fazer *dimming* em curva logarítmica entre 0,1% e 100% que corresponde à sensibilidade do olho humano, [43]. Permite também gerir cenários de iluminação e obter o estado da instalação (podendo assim validar possíveis avarias). Este sistema permite também a interligação de novos equipamentos na mesma rede sem ser necessário alterar a estrutura da instalação inicial, [44].

O sistema DALI permite ligar no mesmo meio de comunicação até 64 equipamentos, podendo estes estar divididos em 16 grupos. Este sistema consegue ainda gerir 16 cenários, [45]. As ligações deste sistema podem ser feitas através de três topologias, nomeadamente em estrela, em série ou a junção das duas, [45].

Existem várias formas de ligar um sistema DALI, sendo a base um sistema *Master-Slave*. Estes sistemas têm como sua constituição base o exemplo mostrado na Figura 4.15.

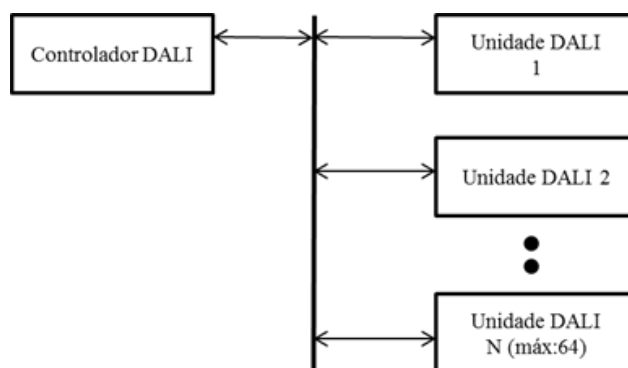


Figura 4.15 - Estrutura da arquitectura Stand Alone de um sistema DALI.

Um sistema DALI poderá estar como *Stand-Alone*, como apresentado na figura anterior, ou poderá fazer parte de um sistema de gestão de edifícios, [45], apresentado na Figura 4.16.

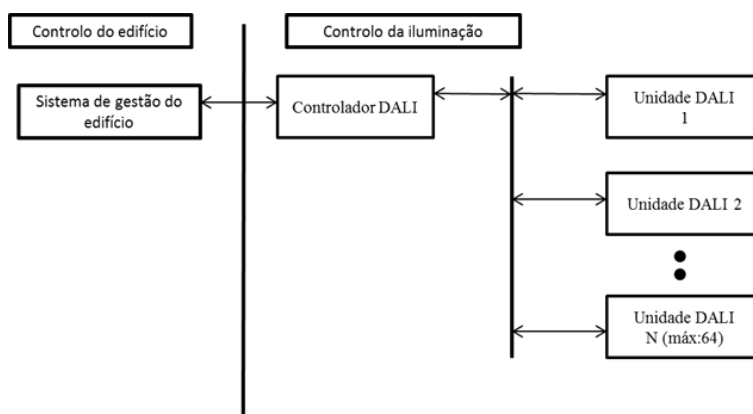


Figura 4.16 - Estrutura da arquitectura de um subsistema *Stand-Alone* de um sistema DALI integrado no sistema de gestão do edifício

Este sistema poderá fazer parte integrante do sistema de gestão de edifícios como mostra a Figura 4.17.

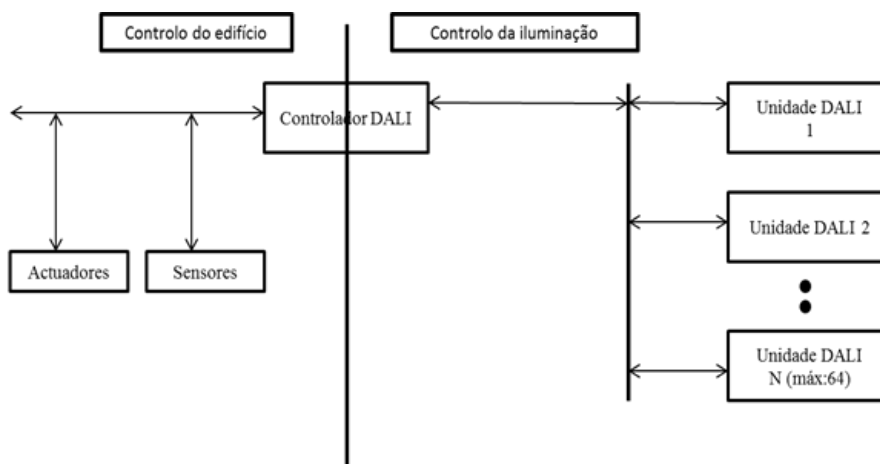


Figura 4.17 - Estrutura da arquitectura de um sistema DALI fazendo parte integrante do sistema de gestão do edifício.

Isto permite que os actuadores, ou sensores, não necessitem de estar ligados directamente ao sistema DALI mas sim ao sistema geral, que por sua vez envia para o *master* do sistema DALI essa informação permitindo assim alargar a área de actuação, [45].

As ligações de um sistema DALI são bastante simples, sendo uma das principais vantagens deste sistema. Na Figura 4.18 é ilustrado o esquema de ligações.

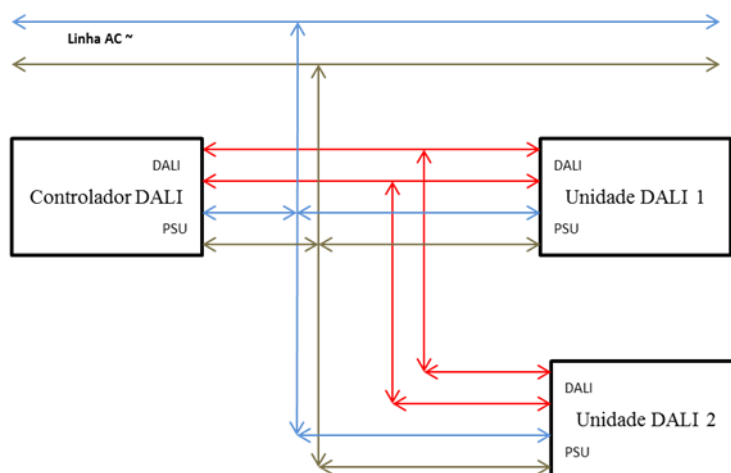


Figura 4.18 - Esquema de ligações de um sistema DALI.

A constituição das ligações consiste somente em dois cabos de alimentação e dois cabos de controlo que torna este sistema de fácil instalação, o que revela uma vantagem. Outra vantagem da ligação de controlo é ser bastante imune a variações ou interferências. Isto porque, porque o nível alto estará entre os 9,5 V e os 22,5 V, sendo o valor típico de 16 V, e o nível baixo estará entre os -4,5 V e os 4,5 V, valor típico 0 V, [45]. As ligações de controlo são ainda imunes a mudanças de polaridade.

Este sistema permite ligações usando cabos com secção 1,5 mm² até 300 m de comprimento máximo se a queda de tensão não exceder os 2 V no final da ligação. Se o comprimento estiver entre os 100 m e os 150 cabos com secção de 0,75 mm² podem ser usados. Se o comprimento da instalação for inferior a 100 a secção do cabo poderá ser de 0,5 mm². Estas ligações não precisam de terminação como o protocolo anterior, o DMX512, [45].

O pacote DALI tem a estrutura apresentada na Figura 4.19.

Endereço	Comando	Dados
----------	---------	-------

Figura 4.19 - Estrutura de um pacote DALI.

A lista de comandos está descrita na referência [46]. A taxa de transferência é de 2400 bits/seg. O endereço poderá ir até 64, ou seja contém 8 bits. Os dados também são formados por 8 bits. É usada a codificação de Manchester, e a sua estrutura prevê a detecção de erros na comunicação.

4.4. Comunicação Zig-Bee

No caso da iluminação uma opção bastante viável é a comunicação usando o protocolo Zig-Bee. Este último é otimizado para ser eficiente tanto ao nível de comunicação como ao nível do consumo energético, [47].

O Zig-Bee é utilizado para criar uma PAN (*personal area network*), utiliza o padrão IEEE 802.15.4 que especifica a camada física e a camada de acesso ao meio. Este protocolo permite comunicar com uma velocidade até 250 kbit/seg, e utiliza as frequências de 868 MHz com um canal utilizado na Europa, 915 MHz com 10 canais com separação entre portadoras de 2 MHz utilizados nos Estados Unidos da América e Austrália, 2,4 GHz em todo o mundo, podendo nesta frequência utilizar até 16 canais com uma separação entre portadoras de 5 MHz, como mostra a figura 4.20, [48].

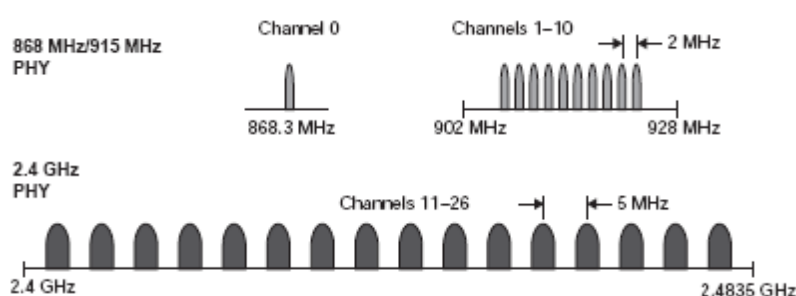


Figura 4.20 - Frequências e canais disponíveis para Zig-Bee.

Este protocolo oferece várias vantagens para ser a escolha para esta aplicação, isto porque é uma comunicação bastante robusta, segura, tem uma boa relação custo/benefício, baixo consumo de energia, [47]. A descrição mais detalhada deste standard é apresentada em anexo.

Principais características

- Baixo consumo e baixo custo

No âmbito desta tese estas duas características, são pontos de escolha bastante significativos. Quando se trata de eficiência energética qualquer poupança de energia pode significar um rendimento do sistema otimizado. No caso do Zig-Bee consegue-se atingir um baixo consumo, visto que as potências de transmissão em causa são baixas e permite que o módulo de comunicação tenha períodos em que não está em funcionamento, ou seja está adormecido. Um módulo de Zig-Bee pode operar com duas baterias tipo AA durante alguns anos sem ser necessário substituir as baterias, [47].

Como o Zig-Bee é um protocolo livre de patentes em que muitos fabricantes oferecem soluções com base neste protocolo, o que permite que se atinja um baixo custo de aquisição de equipamentos, [48]. Outra vantagem será que os módulos de Zig-Bee fornecem não só a capacidade de comunicar mas também são munidos de portas de entrada/saída que podem fornecer várias utilidades, [49].

- Fiabilidade e robustez

A fiabilidade deste protocolo é garantida com o uso de O-QPSK (*Offset-Quadrature Phase Shift Keying*) e DSSS (*Direct Sequence Spread Spectrum*) que fazem com que este protocolo atinja uma boa relação sinal-ruído, [47]. Utiliza também CSMA-CA (*Carrier Sense Multiple Access Collision Avoidance*) que previne colisões no acesso ao meio, [47]. Para garantir que a informação chega ao receptor na sua totalidade e sem erros, um CRC (*Cyclic Redundancy Check*) de 16 bits é usado em cada trama enviada, [47]. Permite efectuar o reenvio até três vezes da mesma trama se existir falhas nos envios anteriores. Se nenhum dos envios tiver sucesso, o receptor comunica ao emissor, que por sua vez permite ao controlador do emissor perceber que a mensagem não chegou ao seu destinatário, [47]. Permite também a definição por qual caminho a trama vai seguir, em topologias *Mesh*. Este ponto será discutido com mais detalhe no ponto do *Routing*.

- Segurança e ritmo de comunicação

A comunicação Zig-Bee é bastante segura, porque usa o standard AES-128 (*Advanced Encryption Standard* a 128 bits). Isto permite que a comunicação seja bastante imune a intrusões o que garante também a robustez e fiabilidade da rede.

Como foi dito anteriormente, este protocolo permite velocidades de transmissão até 250Kbit/seg., mas como a comunicação é Half-Duplex, tem encriptação e existe verificação de recepção o ritmo de comunicação do Zig-Bee é tipicamente de 25 kbit/seg.

5. Projecto de um sistema de iluminação LED inteligente

A luminária proposta para implementar no âmbito desta tese, tem como objectivo apresentar uma elevada eficiência energética recorrendo à tecnologia LED e à utilização de um sistema de gestão inteligente da mesma, incluindo um sistema de comunicação para permitir a cooperação entre luminárias.

A luminária proposta poderá ser aplicada em substituição de luminárias circulares que usem como fonte de luz lâmpadas CFL, tipicamente usadas em escritórios, espaços interiores, corredores, lobbys de entrada. O dimensionamento desta luminária teve em conta também o facto de ser “Plug&Play”, facilitando assim a sua instalação na substituição de luminárias já existentes, não necessitando de cablagem extra relativamente à já existente. A única limitação que esta luminária poderá ter, relativamente às convencionais, é que necessita de uma programação inicial de configuração

5.1. Descrição geral do sistema

5.1.1. Unidade de LEDs

O projecto inicia-se pela selecção do tipo de LED a usar, tendo a escolha recaído no LED OSRON SSL 80 da Osram [51]. As suas características, quer ao nível da temperatura de cor bem como ao nível de luminosidade, permitem que possa ser utilizado em diversas aplicações. Este LED dispõe de uma gama de corrente de funcionamento bastante alargada, entre os 100mA até os 800mA. A temperatura de cor escolhida foi 5000 K, que é uma temperatura de cor bastante usada nas aplicações a que se destina este tipo de luminárias. Outro factor importante é que este tipo de LED atinge um nível de CRI superior a 80, tornando-o adequado para o tipo de aplicação desejada, [51].

A topologia proposta consiste na utilização de 10 LEDs ligados em serie com uma corrente de 470mA. O esquemático que representa as ligações é apresentado na figura 5.1,

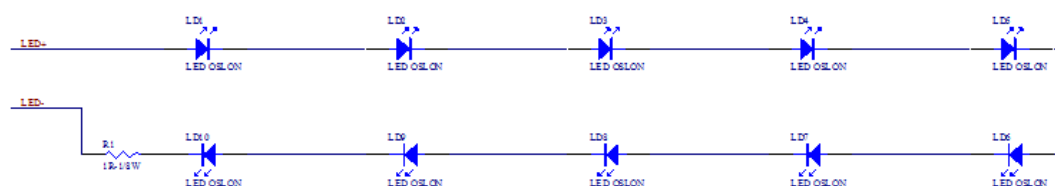


Figura 5.1 - Esquema da placa de LEDs.

Com os 10 LEDs aplicados em série, tensão de alimentação deverá situar-se entre 28 V e 32,5 V, visto que a tensão de cada LED os 2,8 V e os 3,25 V. A queda de tensão esperada em cada LED ronda os 3,1V conforme dados pela ficha de especificação do fabricante. Outro aspecto determinante na escolha da quantidade de LEDs é a quantidade de luz pretendida. Cada LED a 470 mA atinge cerca de 140 lm, visto que para a corrente Obtém-se cerca de 1,25 de relação para o fluxo a 350 mA (podendo

variar entre os 112 lm e os 140 lm conforme gráfico mostrado na figura 5.2 relativo à relação de fluxo *versus* corrente de referência).

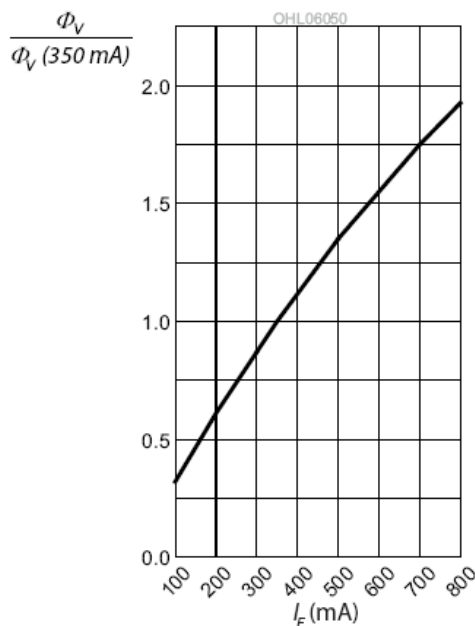


Figura 5.2 - Gráfico da relação entre a corrente e o fluxo luminoso a 350 mA [51].

Com isto podemos esperar que a placa de LEDs consiga atingir os 1400 lm de fluxo luminoso o que faz com que seja comparável a luminárias convencionais para o mesmo tipo de aplicação, mas com uma maior eficiência energética.

A potência estimada para a unidade de LEDs é de,

$$P = VI = 31 \cdot 0,470 = 14,6W \quad (5.1)$$

Outros dispositivos aplicados na placa de LEDs são os sensores de temperatura e o sensor de luminosidade. O sensor de temperatura terá como finalidade monitorar a temperatura de funcionamento dos LEDs evitando assim o desgaste prematuro ou falha de algum LED. O sensor de luz terá como finalidade monitorar a luminosidade do espaço onde a luminária está aplicada.

Sensor de luz

O sensor de luz usado é o SFH3710 da Osram [52]. Este sensor permite detectar o nível de luminosidade que vai desde os 10 lx até os 1000 lx, para uma gama de comprimentos de onda que vai desde os 350 nm até os 1100 nm com um pico de sensibilidade aos 570 nm [52]. Esta característica torna este sensor ideal para o uso em detecção de luz ambiente visto que o pico de sensibilidade do olho humano é bastante semelhante à deste sensor.

O circuito típico par ao uso do sensor e apresentado na Figura 5.3,

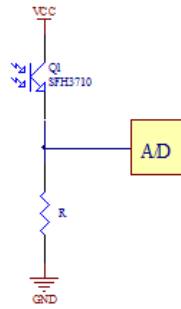


Figura 5.3 - Circuito típico usando o SFH3710.

A luz incidente origina uma corrente proporcional ao fluxo de fótons recebido. Esta corrente é convertida em tensão através da resistência. Esta tensão é digitalizada pelo conversor analógico-digital (A/D) permitindo que seja calculado digitalmente o nível luminoso a que o sensor está exposto.

Para detectarmos com a maior precisão possível, temos de calcular a resistência representada no circuito. Para isso o fabricante fornece-nos a informação que para uma intensidade luminosa de 1000 lx, temos que o sensor atinge uma corrente máxima de 0,5 mA. Por outro lado, a gama dinâmica da entrada em tensão do A/D é de 5 V resultando no valor da resistência dada pela Eq. 5.2.

$$R = \frac{5}{0,0005} = 10k\Omega \quad (5.2)$$

Sensor de temperatura

No caso do sensor de temperatura foi usado o LM335 [53]. Este sensor permite detectar temperaturas que podem ir desde os 233 °K até os 373 °K com uma resolução nominal de 10mV/°K.

O circuito típico de utilização deste sensor é apresentado na Figura 5.4.

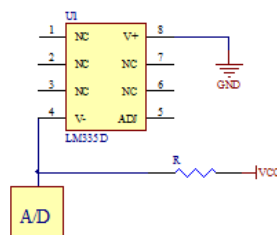


Figura 5.4 - Circuito típico usando o LM335.

Este sensor tem uma queda de tensão máxima de aproximadamente de 3V [53] o que quer dizer que na resistência temos uma queda de tensão de 2V. Se tivermos uma corrente de operação de 1 mA temos o valor da resistência será de,

$$R = \frac{2}{0,001} = 2k\Omega \quad (5.3)$$

A resistência comercial utilizada será a de $2,2k\Omega$ 1/4W.

5.1.2. Dimensionamento do driver

Para controlar a cadeia de LEDs que foi referida no ponto anterior foi necessário dimensionar um *driver*. Conforme referido anteriormente, existem várias topologias para controlar os LEDs. Neste caso optou-se por usar um driver com uma topologia *Buck*, pelas razões apontadas no capítulo 3.

As especificações para o cálculo do driver requerem a informação da tensão de entrada, da tensão e da corrente de saída, e a eficiência esperada. Apresenta-se estas especificações na Tabela 5.1.

	Mínimo	Nominal	Máxima
Tensão de entrada (RMS) [V]	190	230	260
Tensão de entrada (Pico) [V]	260,70	325,27	367,70
Tensão de saída [V]	28	31	32,5
Corrente de saída (ripple de 15%) [mA]	434,75	470	505,25
Eficiência [%]	-	90	-

Tabela 5.1 - Especificações de dimensionamento do driver

O *driver* proposto tem como base o uso de um circuito integrado da Supertex Inc. o HV9910b [54]. Este circuito integrado apresenta várias vantagens relativamente a outros disponíveis no mercado, como por exemplo a simplicidade de utilização e respectivo dimensionamento, custo reduzido, permitir uma gama de tensão de alimentação dos 8V aos 450V e, finalmente permitir dimerização tanto por tensão como por PWM, [54].

Este *driver* permite que o seu dimensionamento tenha como base dois modos de operação, ou por frequência constante ou por tempo a *off* constante.

Na Figura 5.5 apresenta-se os dois circuitos típicos de utilização deste circuito integrado.

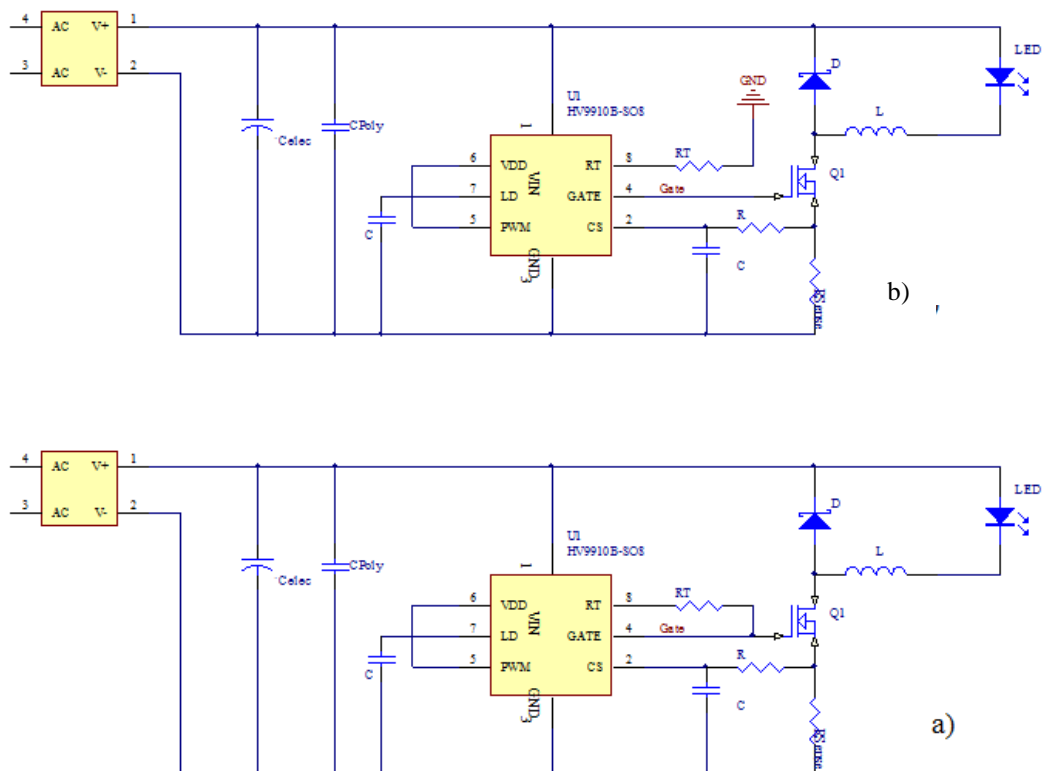


Figura 5.5 - Circuitos típicos usando o HV9910b.
b) Modo de frequência constante. a) Modo de tempo a *off* constante.

No caso da aplicação pretendida, como temos a tensão de entrada a variar a melhor opção será a do tempo a “*off*” constante (Figura 5.5 a). Isto porque podemos ter um tempo a “*on*” variável o que permite uma melhor estabilização da saída, consoante as variações que podemos ter na carga. O HV9910b tem um oscilador interno que permite ajustar a sua frequência de funcionamento ou o seu T_{off} constante, bastando para isso dimensionar uma resistência externa ao HV9910b. Este oscilador consiste num condensador interno de 22 pF, [55], e na resistência R_t a dimensionar.

Este circuito integrado permite que a sua frequência máxima de funcionamento seja de 300 khz, mas quanto mais baixa for a frequência de funcionamento, melhor rendimento é possível obter do driver (as perdas aumentam com a frequência).

Cálculo de R_T

Para este caso vamos optar por o uso de uma frequência nominal de funcionamento que ronda os 80 khz, o que faz com que as perdas de comutação sejam reduzidas e o *driver* tenha o funcionamento correcto. Outro pormenor que será visto em baixo, é que a esta frequência teremos perdas reduzidas. O período para uma frequência nominal de 80KHz é de

$$T_{nom} = \frac{1}{f} = \frac{1}{80000} = 12,5\mu s \quad (5.4)$$

e o duty cycle nominal para o nosso driver é de

$$D_{nom} = \frac{V_{out\ nom}}{V_{in\ nom}} = \frac{31}{325,27} = 9,52\% \quad (5.5)$$

o que dá um $t_{on\ nom}$ de $1,19\mu s$ e um t_{off} de $11,31\mu s$. Então a resistência RT calculada para este T_{off} é de

$$RT = (T_{off}(\mu s) \cdot 25) - 22 = (11,31 \cdot 25) - 22 = 260,75\ k\Omega \quad (5.6)$$

A resistência comercial usada no driver foi de $270k\Omega\ 1/4W$, o que dá um t_{off} resultante de

$$T_{off} = \frac{RT(k\Omega) - 22}{25} = \frac{270 - 22}{25} = 9,92\mu s \quad (5.7)$$

O que faz com que T_{on} seja de aproximadamente de $1\mu s$. Este valor de T_{on} é bastante importante, visto que o HV9910b tem a limitação de não poder ter um este valor abaixo dos 300ns. Isto porque quando o pino da gate passa a HIGH, ou seja o transistor é ligado, isto provoca um pico de corrente que poderia atingir o valor máximo de corrente. Isto poderia provocar uma má leitura da corrente e fazia com que o transistor fosse desligado por se detectar corrente com o valor programado. Para evitar isso, o comparador tem uma latência que vai desde os 150ns a 280ns. Devido a isto é que o $T_{on\ min}$ tem de ser superior a 300ns [56] para que a leitura da corrente seja feita nas devidas condições.

Relativamente à nova frequência de funcionamento nominal com o novo t_{off} ficamos com

$$F_{nom} = \frac{1 - \frac{V_{o\ nom}}{V_{in\ nom}}}{t_{off}} = \frac{1 - \frac{31}{325,27}}{9,92\mu} = 91,2\ kHz \quad (5.8)$$

Mas como temos variação de tensão de entrada e possibilidade da variação de tensão na carga, e como temos um t_{off} constante, temos uma frequência mínima e máxima de operação. Os seus valores são calculados através das expressões 5.9 e 5.10.

$$F_{min} = \frac{1 - \frac{V_{o\ máx}}{V_{in\ min}}}{t_{off}} = \frac{1 - \frac{32,5}{367,70}}{9,92\mu} = 88,61\ KHz \quad (5.9)$$

$$F_{máx} = \frac{1 - \frac{V_{o\ min}}{V_{in\ máx}}}{t_{off}} = \frac{1 - \frac{28}{268,70}}{9,92\mu} = 93,13\ kHz \quad (5.10)$$

Cálculo da bobine L

Outro componente a dimensionar é a bobine. Para a sua determinação temos de ter em conta a tensão máxima da carga ($V_{o\ Max}$), o tempo de descarga (dt) que será t_{off} , e a variação de corrente que a bobine vai sofrer durante a descarga (di).

Partindo destas especificações obtém-se,

$$L = V_{o\text{ máx}} \cdot \frac{dt}{di} = V_{o\text{ máx}} \frac{t_{off}}{\%_{ripple} \cdot I_{o\text{ máx}}} = 32,5 \frac{9,92\mu}{0,15 \cdot 0,50524} = 4,254\text{ mH} \quad (5.11)$$

A bobine escolhida foi a série de três bobines de 1,5 mH, a 0,74A de corrente média, com baixa resistência o que faz com que haja poucas perdas neste componente.

Cálculo de R_{cs}

Para o HV9910b controlar a corrente tem de fazer a leitura da corrente. Para isso este circuito integrado tem um comparador interno que faz a comparação entre 250mV [57], tensão regulada internamente e o pino de CS (Current Sense). A este pino será acoplada uma resistência que será percorrida pela corrente que está a passar nos LEDs quando o transístor está a conduzir. Dada a formação de *ripple* nesta corrente interessa que a comparação seja feita à corrente de pico do *ripple*. Ou seja, a corrente que queremos ler é a corrente de 505,25mA, e como a tensão de comparação é de 250mV ficamos com

$$R_{cs} = \frac{V_{cs}}{I_{pico}} = \frac{0,25}{505,25} = 0,495\Omega \quad (5.12)$$

As resistências aconselhadas, são resistências de 1Ω 1/4W ligadas em paralelo que tem como resistência resultante 0,5 Ω. Devido ao que foi referido relativamente ao $T_{on\text{ min}}$ de ser superior a 300ns, esta tolerância pode não ser suficiente. Para solucionar esta questão podemos complementar o circuito, implementando um filtro RC no pino CS, para que os picos de corrente devido ao ligar do transístor não sejam detectados. O uso de uma resistência e de um condensador provoca um atraso, neste caso 100ns, o que reduz a probabilidade de existir uma leitura incorrecta da corrente. Teremos de ter em atenção de não provocar um atraso demasiadamente elevado, isto para não provocar demasiados atrasos na leitura da corrente. Se o atraso for demasiado podemos atingir picos de corrente elevados o que pode provocar o aumento da corrente RMS. Para compensar o atraso provocado pelo filtro tomou-se a opção de utilizar resistências de 1R1Ω 1/4W em paralelo, como resultante temos 0,55Ω, para que a detecção da corrente seja feita com um valor de pico mais baixo.

Escolha do transístor

A tensão a que o transístor estará sujeito será praticamente a tensão máxima de entrada. Dando uma tolerância de 50% temos que V_{ds} máximo do transístor tem de ser igual ou superior a 551,5V.

Outro aspecto importante é a corrente vai atravessar o transístor. A corrente que atravessa o transístor é a corrente de pico do ripple, ou seja, 505,25mA. Será ideal escolher um transístor que tenha uma

corrente rms elevada que corrente limite visto que assim consegue-se ter um R_{ds} do transístor será mais baixo.

Outra limitação da escolha do transístor é a restrição dada pelo fabricante relativamente à carga eléctrica da *gate* do transístor tem de ser inferior a 25 nC para frequências de funcionamento inferiores a 100 kHz e 15 nC se para frequências de funcionamento superiores a 100 kHz. Isto porque a corrente que o circuito integrado consome depende da carga da *gate* do transístor escolhido. Esta corrente depende também da frequência de funcionamento do circuito. Então a corrente que o integrado consome é

$$I_{circ.int.} = 1.0mA + Q_g \cdot F_s \quad (5.13)$$

onde 1,0 mA é a corrente que os circuitos internos do HV9910b consomem. Tomando o exemplo de um transístor com uma carga de *gate* com 25 nC [54] e tivermos uma frequência de operação de 120 kHz, temos que a corrente consumida pelo HV9910b é de

$$I_{circ.int.} = 1.0mA + 25nC \cdot 120KHz = 4mA \quad (5.14)$$

Sabendo que o HV9910b pode ser alimentado com uma tensão máxima de 450 V temos que a potencia dissipada no é de 1,8 W. Embora esta potência seja somente dissipada quando o transístor está a ligado, ou seja o pino de controlo da *gate* está a HIGH, temos de ter em conta visto que o objectivo é obter um driver que tenha as menores perdas possíveis temos que a capacidade de dissipação do HV9910b tem de ser tida em conta.

O transístor escolhido é o FQPF6N60C da Fairchild, apresenta uma carga de *gate* de 16 nC e um $R_{ds\ on}$ de 2 Ω . Este transístor apresenta um bom compromisso entre a sua capacidade de *gate* e a sua resistência de condução, visto que apresenta uma baixa resistência de condução o que minimiza perdas no transístor, e uma capacidade dentro do aconselhado pelo fabricante do HV9910b o que minimiza as perdas no integrado.

Em termos de potência dissipada no transístor quando este está ligado, ou seja em T_{on} com a corrente média de 470 mA, este dissipa 0,442 W durante esse período. As perdas devido à capacidade de *gate* no circuito integrado devido a este transístor, temos que a tensão máxima de alimentação será de 367,7 V, a frequência máxima de funcionamento será de 93,13 kHz a potência máxima dissipada no circuito integrado será de 0,9 W, visto que a corrente que ele consome é de 2,5 mA.

Outra característica do transístor é o facto de o seu V_{ds} máximo é de 600 V, o que está acima dos 551,5 V aconselhados.

Escolha do diodo

Visto que o diodo estará ligado directamente a alimentação, que pode atingir um valor máximo de 367,70 V, temos de escolher um diodo que tenha uma tensão inversa superior a esta. Para dar uma

tolerância de 50%, como foi feito no transistor, a escolha do diodo será feita tendo em conta que a tensão inversa seja maior a 551,5 V.

Como a corrente de pico do *ripple* é de 505,25 mA, teremos de escolher um diodo que suporte essa corrente. Existem dois diodos da Diotec que satisfazem estas condições, o modelo ES1J e o ES2J. O primeiro, ES1J, tem uma tensão inversa de 600 V e suporta uma corrente directa de 1A. O segundo, o ES2J, tem uma tensão inversa de 600 V e suporta uma corrente de 2 A.

O diodo escolhido foi o ES2J, isto porque a queda de tensão à corrente de operação do driver inferior a 1,1 V enquanto o ES1J tem uma queda de tensão à corrente de operação de aproximadamente de 1,2 V. Com isto minimizamos a potência dissipada no diodo.

Ponte rectificadora

Como temos uma tensão de alimentação alternada, necessitamos de fazer com que a tensão de alimentação fique somente positiva. Para isso rectificamos a onda de entrada usando uma ponte rectificadora.

A ponte rectificadora escolhida é a S380F da Diotec que suporta uma tensão RMS constante de 380 V, e uma corrente de 0,8 A. Como temos uma tensão de alimentação máxima de 260 V esta ponte respeita esta especificação. Relativamente à corrente a corrente esperada que o circuito poderá consumir será de

$$I_{\text{máx.ponte}} = \frac{V_{o \text{ máx}} \cdot I_{o \text{ máx}}}{V_{\text{min DC}} \cdot \eta} = \frac{32,5 \cdot 505,25 \times 10^{-3}}{65 \cdot 0,9} = 280,7 \text{ mA} \quad (5.15)$$

onde $V_{\text{min,DC}}$ é o dobro da tensão máxima de saída, e η a eficiência esperada.

Antes da ponte foi usado um TVS (Transient Voltage Suppression) de 400 V para suprimir picos acima de 400 V que possam surgir da rede de alimentação, protegendo assim o circuito.

Condensadores de entrada

Para estabilizar a tensão de entrada, será necessário utilizar uma capacidade. A capacidade mínima a utilizar é dada por

$$\begin{aligned} C_{\text{min.entrada}} &= \frac{V_{o \text{ máx}} \cdot I_{o \text{ máx}}}{(2 \cdot V_{\text{min,ac}}^2 - V_{\text{min,dc}}^2) \cdot \eta \cdot \text{freq}} \\ &= \frac{32,5V \cdot 505,25mA}{(2 \cdot 190^2 - 65^2) \cdot 0,9 \cdot 91,2KHz} = 2,81 \mu F \end{aligned} \quad (5.16)$$

Então a capacidade de entrada total terá de ser superior ao valor calculado. Para implementar um circuito de PFC, usa-se o circuito de Valley-Fill e o valor dos condensadores é pelo menos o dobro da capacidade mínima. A tensão dos condensadores usados é de 400 V e a capacidade é de 4,7 μF .

Circuito resultante

5.2. Projecto do sistema de controlo

O funcionamento da luminária proposta consiste em ter um sistema que cubra uma área e que a luminosidade seja somente imposta onde necessário e quando necessário. Um exemplo será um *open-space* onde a utilização não é constante e somente em algumas parcelas do mesmo não é necessário ter a iluminação a funcionar a 100%.

Outra consideração que foi tida em conta foi a luminosidade do espaço. Ou seja, as luminárias só necessitam entrar em funcionamento se a luminosidade estiver abaixo de um certo valor de referência, por exemplo ao anoitecer. Isto será implementado pelo sistema de gestão com o auxílio do sensor de luz que foi implementado na placa de LEDs como descrito anteriormente.

Então a solução proposta será, nos locais em que a luminosidade seja inferior ao valor de referência e que estejam com ocupação real a iluminação esteja com o fluxo luminoso a 100%, na área circundante a 75% e nas restantes zonas a 50%, somente para dar iluminação ambiente. Para implementar esta solução foi necessário desenvolver um sistema de gestão e um sistema de comunicação. Estes sistemas serão descritos nos pontos seguintes.

Para dar um exemplo de aplicação deste sistema, a figura 5.7 mostra uma situação de acção deste sistema.

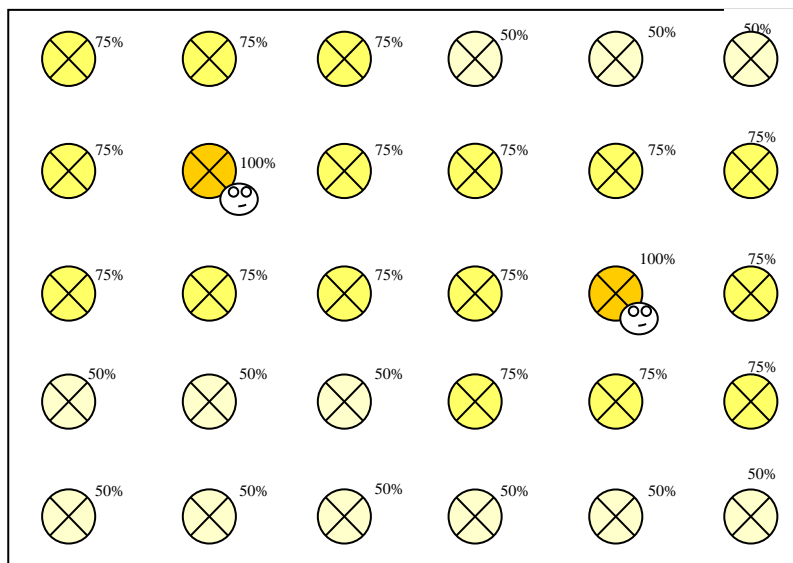


Figura 5.7 - Exemplo de uma situação estado possível das luminárias em um espaço.

Como podemos observar pela figura anterior, vemos que quando a área que está por debaixo da luminária está ocupada esta fica a 100% de intensidade, as luminárias circundantes ficam a 75% para complementar a luminosidade na área ocupada e as restantes ficam a 50% para dar uma luz ambiente e não fiquem completamente escuras.

Para dar um pequeno exemplo, se assumirmos que a luminária tem um comportamento linear relativamente ao consumo de potência e esta a 100% consumir 20W, se tivessem todas ligadas a 100% sem interacção entre elas este sistema consumiria na totalidade 600w, visto que temos 30 luminárias.

Para o caso de haver interacção, temos pela tabela 5.2 os seguintes valores

Percentagem de intensidade	Potência [w]	Nº de luminárias	Potencia Consumida [W]
100%	20	2	40
75%	15	16	240
50%	10	12	120
Potencia total consumida			400

Tabela 5.2 - Exemplo de valores de consumo com interacção entre luminárias

Para o caso representado na figura temos que o fluxo luminoso a 50% a luminária consome 10 W e a 75% consome 15 W, para este caso a potência consumida da instalação será de 400 W o que representa uma poupança de 33,33%. Isto pode significar em termos de poupança financeira uma quantia elevada para espaços muito grandes, ou com pouca utilização.

Uma análise mais exaustiva e comparativa entre tecnologias de luminárias será feita no capítulo de demonstração de resultados.

5.2.1. Unidade de controlo da luminária

Para implementar o sistema de gestão da luminária é proposto o uso de um micro controlador que fará a leitura dos sensores acoplados e efectuará acções consoante as variáveis exteriores. Para implementação do protótipo foi utilizado uma placa de desenvolvimento, o Arduino UNO.



Figura 5.8 - Placa de desenvolvimento Arduino UNO.

Esta placa tem como base um micro controlador ATMEGA328P que tem um clock de 16 MHz, tem 6 entradas analógicas com conversores A/D com resolução de 10bits, tem 14 entradas/saídas digitais podendo 6 delas funcionar como saídas de PWM, permite a ligação USB através de ATMEGA16U2, tem uma EEPROM de 1 kb, uma SRAM de 1kb e uma memória flash de 32 kb sendo 0,5 kb reservados para o bootloader. Esta placa permite várias funcionalidades, sendo elas adequadas para a realização deste protótipo.

O uso de um módulo de comunicação, será necessário para implementar as funções desejadas por este sistema de gestão. O esquema de ligações simplificado para a interacção entre luminárias é o seguinte



Figura 5.9 - Esquema de ligação para interacção entre luminárias.

O módulo de comunicação será falado no ponto seguinte, mas a interacção entre o módulo de gestão (Arduino) e a placa de driver será feito através de uma porta com saída em PWM disponível no Arduino. Esta saída será ligada a um foto acoplador para fazer o isolamento galvânico entre o Arduino e a placa de driver. O circuito associado a este foto acoplador está representado na figura 5.10.

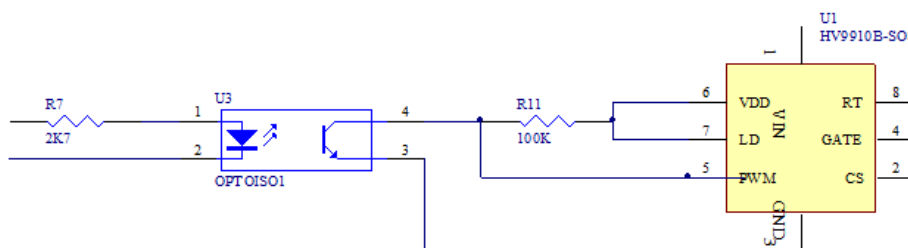


Figura 5.10 - Esquema de ligação da saída de PWM ao driver.

Este circuito faz com que o PWM fique invertido, ou seja, quando o díodo do foto acoplador está a conduzir o transistor impõe GND ao pino de PWM do HV9910b. Quando temos o díodo a não conduzir é imposto 5V. O que quer dizer que quando temos um PWM com um duty cycle de 0 temos o HV9910b ao máximo, quando temos o PWM com um duty cycle máximo o HV9910b está no mínimo.

Modos de operação

O sistema de gestão implementado no Arduino UNO tem duas vertentes, uma de gestão autónoma e a outra a gestão dependente das luminárias que estão ao seu redor. Para isso foi criado o conceito de modo de funcionamento, que traduz a estado actual da luminária. Na tabela 5.3 é descrito os modos de funcionamento possíveis.

Modo	Função	Resultado na luminosidade
WAIT	Modo de repouso da luminária	Fluxo luminoso a 0%
TRACTUS	Modo accionado por movimento	Fluxo luminoso a 100%
SUN	Modo accionado por detecção de pouca luminosidade do ambiente em redor à luminária	Fluxo luminoso a 50%
TRACTUS_AUX	Modo accionado por uma luminária circundante	Fluxo luminoso a 75%
PROTECT	Modo de protecção accionado por excesso de temperatura	Fluxo luminoso a 0%

Tabela 5.3 - Modos de funcionamento da luminária.

No modo WAIT a luminária está em repouso, ou seja, a luminosidade do espaço está acima do valor de referência, não existe nenhum movimento, nem foi recebida nenhuma informação das luminárias circundantes. A acção que o modulo de gestão toma é pôr o fluxo luminoso a 0%.

No modo TRACTUS a luminária fica com o fluxo a 100%. Isto é resultante de o sensor ter detectado movimento, indicando assim presença no raio de acção da luminária. Quando a luminária entra neste modo envia através do módulo de comunicação a informação que está a funcionar neste modo para as luminárias circundantes agirem em conformidade com o previsto.

No modo SUN a luminária fica com o fluxo luminoso a 50%. Isto é resultado da leitura do sensor de luz, indicar que a luminosidade do espaço está abaixo do valor de referência.

No modo TRACTUS_AUX a luminária fica com o fluxo a 75%. Isto é resultado da recepção através do módulo de comunicação que uma luminária circundante a esta entrou no modo TRACTUS. Assim a luminária fica como auxiliar da luminária que entrou em modo TRACTUS.

No modo PROTECT a luminária apagar-se-á por protecção. Esta protecção é uma protecção de temperatura. Quando é feita a leitura do sensor de temperatura, e esta leitura está acima do valor máximo de referência esta apaga-se. Isto servirá para reduzir os danos nos próprios LEDs para que a fiabilidade dos mesmos não seja posta em causa.

Algoritmo de operação

Na figura 5.11 é representado o diagrama que mostra a sequência de funcionamento da luminária



Figura 5.11 - Diagrama de funcionamento normal da luminária proposta.

Quando a luminária é ligada, esta entra um modo curto de configuração. Isto porque o funcionamento da luminária depende de comparações com valores de referência e de luminárias circundantes e de um módulo de comunicação. Este módulo de comunicação é interligado através da porta série do micro controlador, para isso uma configuração é necessária do baud rate da porta é necessário definir. Para esta aplicação foi definido o valor de 38400 de baud rate. Com este baud rate a fiabilidade e rapidez da comunicação é garantida. Outra configuração necessária é a dos valores de referência dos sensores. Estes valores encontram-se gravados previamente na EEPROM do micro controlador, nas posições 1 e

2, e são passados para duas variáveis a *max_val_sun* que indica o valor de máximo de luminosidade e o *max_val_temp* que indica o valor máximo de temperatura permitida. Em seguida será necessário fazer a mesma operação para as luminárias circundantes. Ou seja, estes valores estão guardados na EEPROM do micro controlador e serão passadas para o vector *luminárias_gestão* que pode conter 20 luminárias circundantes mas que pode ser alterado para um número superior com o limite da capacidade da EEPROM usada. Em seguida a luminária entra em modo WAIT para entrar no ciclo de funcionamento.

Com a luminária em funcionamento, a primeira validação que será feita é se a temperatura da placa de LEDs está dentro do esperado, se não estiver a luminária entra em modo de PROTECT ficando neste modo até a temperatura atingir um valor 10°C inferior à temperatura limite. O cálculo da temperatura usando o sensor de temperatura colocado na placa de LEDs é feito da seguinte forma

$$Temp = (Leitura_Sense / 1023.0 * 5 * 3.0 * 100) - 273.15 \quad (5.17)$$

Em seguida será feita a leitura da luminosidade da área, através do sensor de luz colocado na placa de LEDs. Esta leitura é feita interrompendo por 1 ms, ciclicamente, o fluxo dos LEDs, isto para não fazer uma leitura errada da luminosidade, em seguida é feito cálculo da seguinte forma

$$Luz = Leitura_{sense_{luz}} \times \left(\frac{5}{1024}\right) \quad (5.18)$$

Se o valor for inferior ao da referência a luminária entra em modo de SUN, se for superior a luminária continua em modo de WAIT até que a verificação da luminosidade indique que a luminosidade é inferior.

Se a luminária entrar em modo de SUN só pode acontecer uma de três coisas. Ou mantém-se ou sai do modo SUN por validação de luminosidade, ou pode entrar em modo TRACTUS ou pode ainda entrar em modo TRACTUS_AUX. Se o sensor de movimento ficar activo, a luminária entra em modo TRACTUS e só sairá desse modo se nenhum movimento for detectado. Se não for detectado nenhum movimento será feita a leitura do vector que contém a informação sobre o modo das luminárias circundantes. O vector é constituído com o número da luminária e o seu estado, e tem a seguinte estrutura.

Nº da luminária circundante	Estado
12	0
45	1
● ●	
23	0

Figura 5.12 - Estrutura do vector das luminárias circundantes.

É feita a validação do vector, e se alguma posição tiver a 1 (o que significa que essa luminária está em modo de funcionamento TRACTUS. Na figura a luminária nº45 está em modo TRACTUS.), a luminária entrará em modo TRACTUS_AUX. A luminária manter-se-á neste modo ou até que se detecte movimento, ou se o modo de todas as luminárias circundantes ficar diferente de TRACTUS. Esta informação chega a esta luminária através do módulo de comunicação, como será descrito no ponto seguinte.

O ciclo descrito repetir-se-á infinitamente enquanto a luminária estiver em funcionamento.

A luminária poderá funcionar de forma autónoma, sem módulo de comunicação, visto que somente a funcionalidade de ter em conta o estado das luminárias circundantes não está operacional, mas não será tão eficaz consoante o pretendido. Conforme dito anteriormente este ciclo de funcionamento completo incluído o módulo de comunicação trará uma melhoria em termos de eficiência energética, visto que só quando for necessário é que a luminária estará a fornecer luminosidade máxima o que se traduz em uma poupança significativa de energia.

5.2.2. Unidade de comunicação

O sistema de comunicação terá como base o uso de um módulo de Zig-Bee, mas poderá usar outro módulo de comunicação que tenha uma porta série para comunicar com o módulo de gestão da luminária.

O módulo de comunicação usado para efectuar este protótipo é o módulo XBee Series 2. A figura 5.13 mostra o módulo usado no protótipo



Figura 5.13 - Módulo XBee Series 2 da Digi®.

Este módulo tem 2mW de potência de emissão que permite atingir até 120m de alcance. Para a aplicação em causa é indicado visto que as distâncias em causa são bastante abaixo do limite. Este módulo transmite a uma velocidade máxima de 250Kbps, o que para as transmissões e recepções em causa será adequado.

Configuração da rede

Como foi referenciado em capítulo anterior, para se formar uma rede de Zig-Bee temos de ter pelo menos um coordenador, sendo os restantes módulo configurados como Router/End-Devices. Para configurar o modo de funcionamento da rede, cada módulo necessita de ser programado. Para isso foi usada a ferramenta disponibilizada pela Digi® o X-CTU. Para utilização desta ferramenta foi

necessário configurar a porta série para uma velocidade de 38400. Em seguida foi configurada uma rede em modo AT nesta ferramenta. A 5.14 mostra alguns pontos que foram configurados tanto para o coordenador da rede como para os restantes módulos.

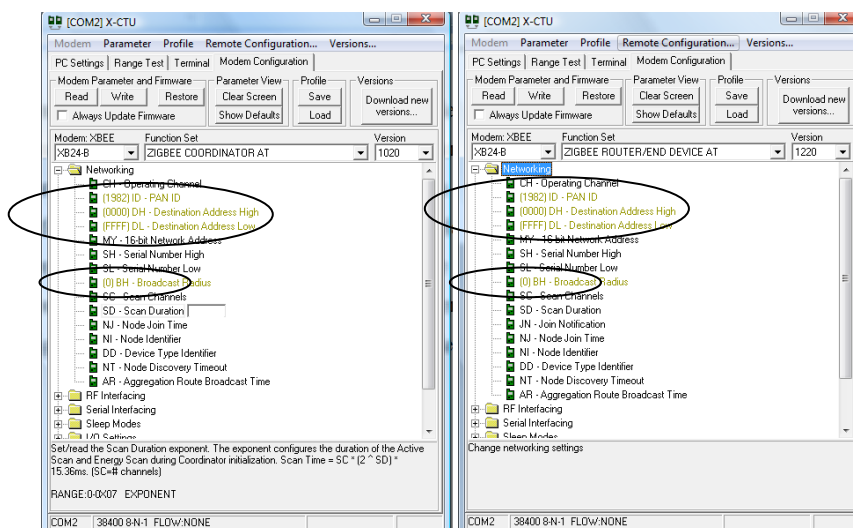


Figura 5.14 - Configuração da rede Zig-Bee no *software* X-CTU.

Para este protótipo o PAN ID foi configurado para ser 1982, e a rede terá o funcionamento em broadcast. Isto porque assim consegue-se que qualquer luminária em redor possa receber os dados, e o tempo de reacção dessas luminárias será idêntico não havendo diferenças momentâneas na iluminação. Para a configuração da rede em broadcast, configurou-se o Destination Address High para 0X0000 e o Destination Address Low para ser 0xFFFF que indica que o módulo irá comunicar em broadcast. Outro parâmetro importante a configurar visto que a rede vai funcionar em broadcast é o raio de alcance. Definiu-se o raio de alcance no máximo, isto para este protótipo, mas em aplicações futuras será importante restringir o raio de acção ao necessário, isto para prevenir conflitos de redes diferentes no mesmo espaço.

A segurança na rede não foi configurada neste protótipo visto não ser necessária a sua aplicação mas em aplicações comerciais fará todo o sentido ser aplicada.

Estrutura da trama

Foi criada uma forma de transmitir os dados para permitir alguma dinâmica no que diz respeito ao tamanho da trama. A trama implementada foi a seguinte

&	Receptora	&	Emissora	&	Código	&	Valor	&	CRC	&
---	-----------	---	----------	---	--------	---	-------	---	-----	---

Figura 5.15 - Estrutura da trama definida.

Esta trama é composta por cinco campos. O primeiro campo será a informação para quem é enviado, o segundo campo quem enviou, o terceiro campo o código a executar, o quarto o valor enviado, e o quinto o CRC. Estes campos foram definidos como valor inteiros inseridos em um vector de caracteres. No protótipo o tamanho máximo de cada campo poderá ser de 20 caracteres, podendo este tamanho ser dinâmico, podendo a trama ter no máximo 106 caracteres ou seja pode ter no máximo 212 bytes.

O dinamismo desta trama é garantido pelo carácter limitador entre campos, o &. Isto permite que o tamanho de cada campo seja variável dentro do limite que está definido. Na figura 5.16 mostra-se duas possibilidades de conteúdo da trama.

&	1	&	54	&	6	&	1234	&	1295	&
&	999	&	54676	&	6	&	8765	&	64446	&

Figura 5.16 - Exemplos de tramas.

Podemos analisar que o tamanho dos campos pode ser variável o que torna também o tamanho da trama variável. Pegando no exemplo e da trama de cima, podemos analisar que a receptora desta trama será a luminária nº1, que quem emitiu foi a luminária nº54, o código a executar será o código nº6, com o valor 1234 e um CRC de 1295.

Este CRC será calculado com base nos restantes valores, visto que os campos somente são preenchidos por valores inteiros. O valor do CRC é a soma de todos os outros valores. Voltando a pegar no exemplo da trama superior vemos que o CRC é dado por,

$$\begin{aligned} CRC &= Receptora + Emissora + Código + Valor \\ &= 1 + 54 + 6 + 1234 = 1295 \end{aligned} \quad (5.19)$$

Apesar da trama de Zig-Bee já conter um CRC e esta verificação já tenha sido feita uma segunda validação com o CRC mostrado na equação anterior, permite que no receptor seja feita a validação se todos os dados chegaram correctos.

Algoritmo de comunicação

A Figura 5.17 mostra a sequência de funcionamento do módulo de comunicação e sua interacção com o funcionamento normal da luminária.

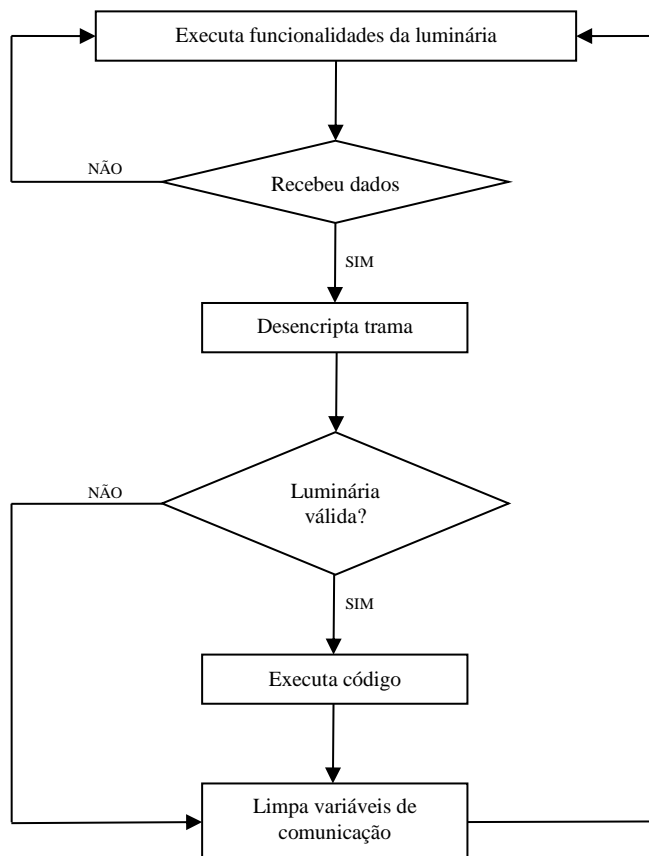


Figura 5.17 - Diagrama de recepção de dados através do módulo de comunicação.

Depois de a luminária executar as suas funcionalidades conforme descrito no ponto acima, antes de reiniciar um novo ciclo, esta vai verificar se existem dados no buffer do modulo da comunicação. Se não existirem então passa para outro ciclo, se existirem esta vai agir em conformidade com o que recebeu.

Depois de se receber os dados do módulo, a trama recebida vem da forma que foi descrita no ponto anterior. Para tornar os dados acessíveis é executada uma função que faz a descriptação da trama para vectores de caracteres. Como esta trama tem a vantagem de ser dinâmica esta descriptação da trama é feita de caracter limitador em carácter limitador. Depois da descriptação feita é validada a variável com a informação da receptora. Se não for para a luminária em questão, esta descarta a informação e elimina a informação da comunicação. Se for a luminária em questão, esta valida o CRC, e se for válido esta vai executar o conteúdo da trama. Esta trama contém em forma de código as funções a desempenhar que serão descritas no ponto seguinte.

Para o caso do envio de dados a sequência está representada na Figura 5.18 .

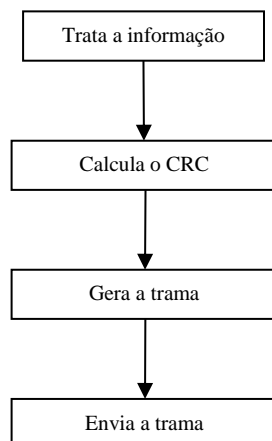


Figura 5.18 - Diagrama de envio de dados através do módulo de comunicação.

Quando a luminária precisa de comunicar, por exemplo quando tem de enviar a informação que entrou em modo TRACTUS, esta prepara os dados para serem enviados, calculando o CRC, gerando a trama colocando os caracteres limitadores e envia a trama. Este envio pode ser efectuado em qualquer altura, bastando para isso enviar através da porta série e o módulo de comunicação é que fará a gestão do envio dos dados. O tempo de execução deste processo é bastante rápido não influenciando o funcionamento da luminária.

Tipos e funções dos códigos usados

Consoante o que será necessário enviar ou receber, esta informação está em forma de tipo de códigos. A tabela 5.4 mostra os códigos propostos para o protótipo.

Código	Função	Tipo
2	Leitura do sensor de luminosidade	Teste
3	Leitura do sensor de temperatura	Teste
4	Leitura do sensor de movimento	Teste
5	Leitura do PWM	Teste
6	Escrita do valor de PWM	Teste
7	Escrita do valor máximo de luminosidade	Programação
8	Escrita do valor máximo de temperatura	Programação
9	Leitura do valor máximo de luminosidade	Teste
10	Leitura do valor máximo de temperatura	Teste
11	Escrita do nº das luminárias circundantes	Programação
12	Leitura do nº das luminárias circundantes	Teste
13	Forçar valor do sensor de luminosidade	Teste
14	Forçar o valor do sensor de temperatura	Teste

Código	Função	Tipo
15	Leitura do modo de funcionamento actual	Teste
16	Activação/Envio do modo TRACTUS_AUX	Normal
17	Desactivação/Envio do modo TRACTUS_AUX	Normal

Tabela 5.4 - Tipos e funções dos códigos das tramas.

Existem três tipos de códigos, uns de teste, uns de programação da luminária e outros de funcionamento corrente da luminária.

Os códigos de teste serão para ser usados por uma ferramenta desenvolvida para o efeito, que será explicada no ponto seguinte. Estes códigos consistem na leitura de sensores, do modo de funcionamento, da leitura da memória, ou escrita forçada de variáveis para estudar o comportamento da luminária. Os códigos de leitura tem como funcionamento fazer a leitura do ou dos sensores (luminosidade, temperatura ou movimento) ou das variáveis (modo, luminárias circundantes ou PWM, valores máximos permitidos) que queremos obter informação e enviar para a ferramenta de teste. O código de escrita do valor do PWM consiste em forçar o valor do PWM, alterando o valor da variável que comanda o PWM para validar a acção da luminária.

Os códigos de programação serão usados para definir as luminárias que estão em redor, definir patamares tanto em luminosidade como em temperatura. Este parâmetros serão definidos dependendo do local a instalar pela luminária, exceptuando o valor da temperatura que só deverá ser alterado se estritamente necessário (como por exemplo para teste). Estes parâmetros serão escritos na memória EEPROM do micro controlador. Estes códigos alteram também as variáveis locais para que seja actualizado o funcionamento da luminária para que não seja necessário reiniciar a luminária, com excepção da programação das luminárias circundantes.

Os códigos de funcionamento normal serão os usados quando a luminária estiver em uso normal. O código 16, que é a activação do TRACTUS_AUX, tem duas funções. A primeira será enviar para as luminárias circundantes um sinal de TRACTUS para que estas regulem o fluxo luminoso e a segunda será quando recebe este tipo de código regule o fluxo da própria luminária para o fluxo associado ao TRACTUS_AUX. Quando recebe este código um vector de estado das luminárias circundantes será actualizado. Por sua vez o código 17 será o inverso, ou seja, quando é enviado pela própria luminária é indicado às luminárias em redor que esta luminária saiu do modo TRACTUS. Quando o recebe esta luminária irá validar se todas as luminárias circundantes não estão em modo TRACTUS, e se assim for ela passa a modo SUN. Daí a importância do uso deste vector de estado das luminárias circundantes, para que não seja alterado o estado da luminária indevidamente.

5.3. Ferramenta de teste e programação

Para implementação e teste da luminária foi desenvolvida uma ferramenta que permite executar algumas funções. Esta ferramenta tem como base uma aplicação desenvolvida em C#, que permite ao utilizador testar e programar uma luminária em tempo real.

A figura 5.19 mostra o esquema de ligações para a utilização desta ferramenta.



Figura 5.19 - Esquema de ligação para a ferramenta programação e teste.

Como temos uma ligação série temos de configurar esta ligação, a ferramenta permite esta configuração na primeira janela da mesma. A Figura 5.20 mostra esse interface.

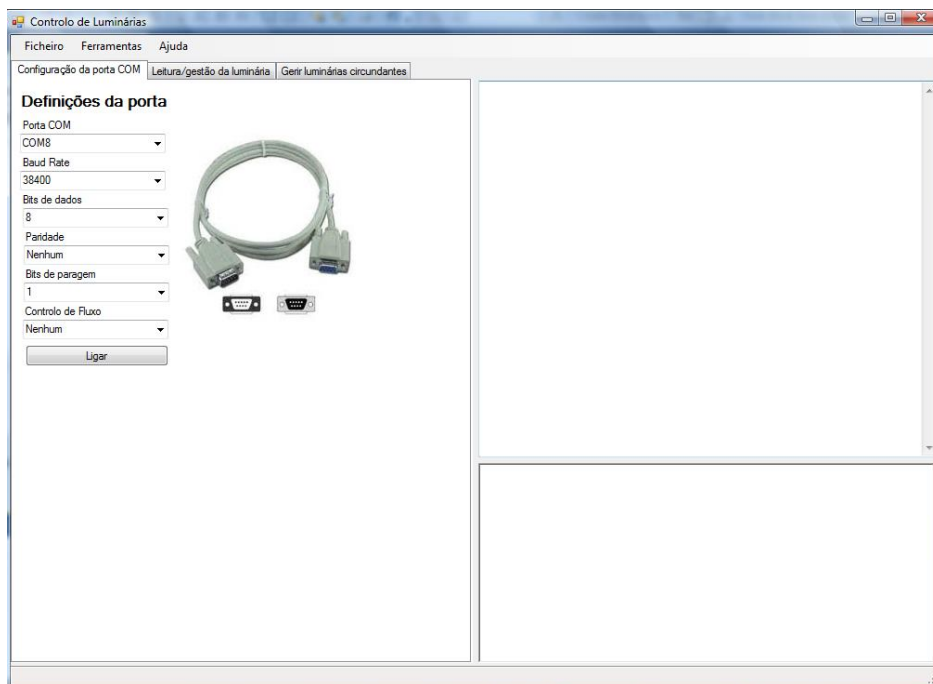


Figura 5.20 - Interface para configuração da porta série.

Depois da configuração feita, uma validação de comunicação é feita. Com esta validação efectuada podemos então passar para o modo de funcionamento da ferramenta. A tabulação Leitura/gestão da luminária permite efectuar apresenta a seguinte interface.

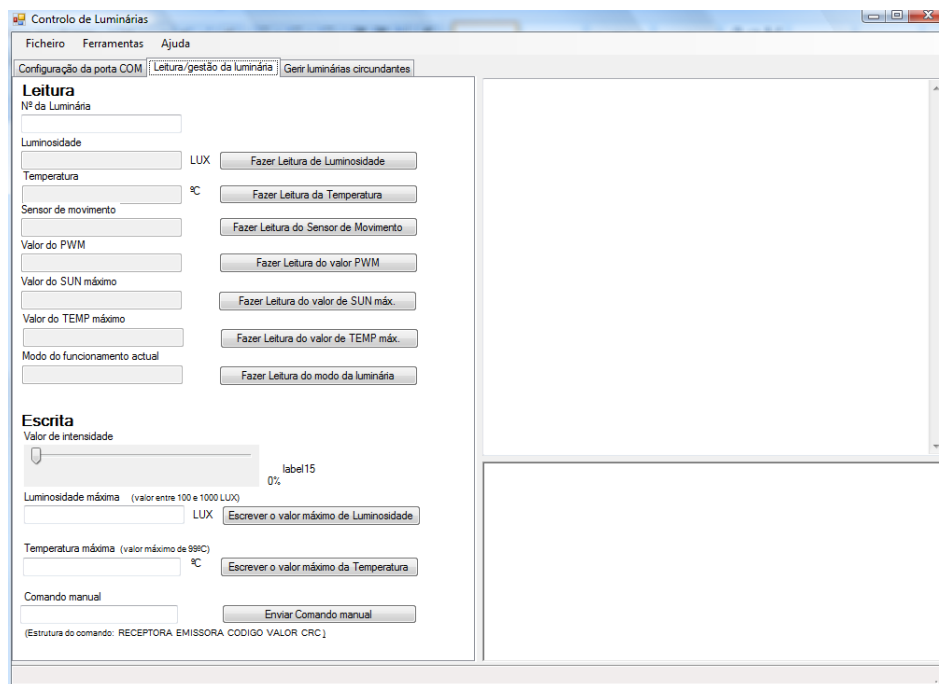


Figura 5.21 - Interface para leitura/gestão da luminária.

Neste interface temos disponíveis as leituras dos sensores, do modo, dos valores máximo permitidos, permite regulação do fluxo luminoso da luminária e permite também o uso de código manual para teste.

Esta aplicação usa o mesmo conceito que foi implementado no micro controlador com o módulo de comunicação. Ou seja, cada vez que um pedido é feito é enviada uma trama com as mesmas características desse pedido e o micro controlador responde da mesma maneira. Em cada comunicação é feita a verificação se foi implementada pelo micro controlador. Basicamente esta aplicação porta-se como uma luminária circundante, de uma forma mais completa, não influenciando o funcionamento.

Na tabulação Gerir luminárias circundantes, podemos como o próprio nome diz gerir as luminárias circundantes ao redor da luminária em questão. A Figura 5.21 mostra o interface para implementação destas funções.

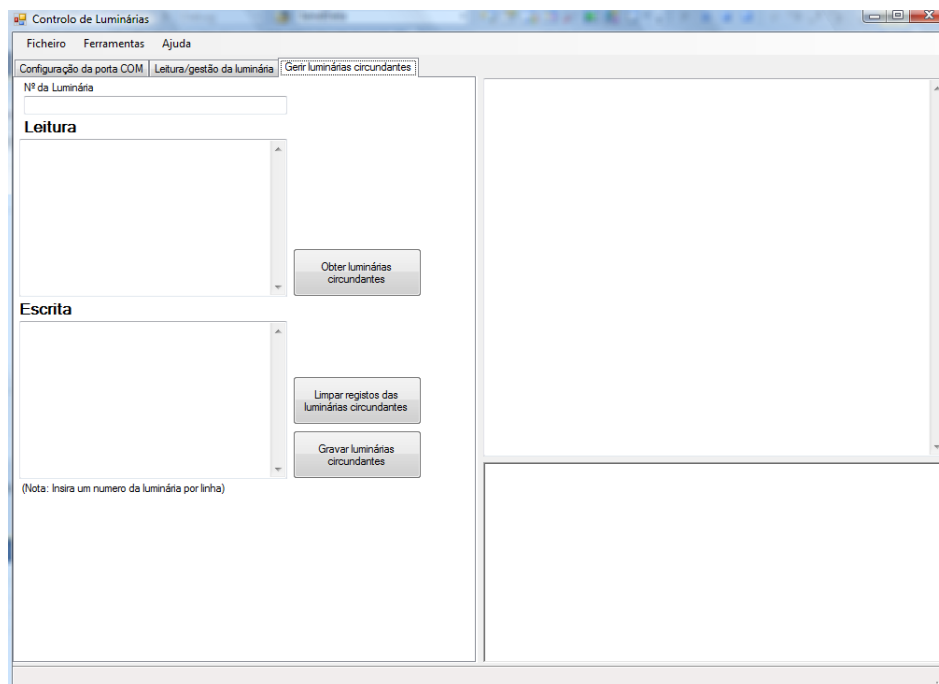


Figura 5.22 - Interface para gestão das luminárias circundantes.

Com a ferramenta podemos fazer a leitura das luminárias circundantes já registadas na luminária. Esta leitura será feita, como dito anteriormente, directamente da EEPROM do micro controlador. Podemos também actualizar a informação, acrescentando novas luminárias, ou fazer o reset à memória do micro controlador onde está guardada esta informação.

Esta ferramenta torna-se bastante útil para efectuar diagnósticos de operação, diagnósticos de funcionamento dos sensores, instalações e reparações nos locais onde a luminária esta aplicada.

6. Ensaio experimental do sistema de iluminação

Neste capítulo será feita a avaliação de resultados obtidos. Para avaliação dos resultados foi necessário efectuar a montagem de um protótipo, efectuar a testes eléctricos, efectuar a teste fotométrico e fazer uma avaliação energética de um sistema com iluminação convencional e com a luminária proposta.

6.1. Montagem do protótipo

Para efectuar o protótipo foi necessário recorrer ao uso de uma ferramenta de desenho de esquemáticos e de placas de circuito impresso. O software usado foi o Altium Designer.

Placa de LEDs

O esquemático desenhado para a placa de LEDs é apresentado na Figura 6.1.

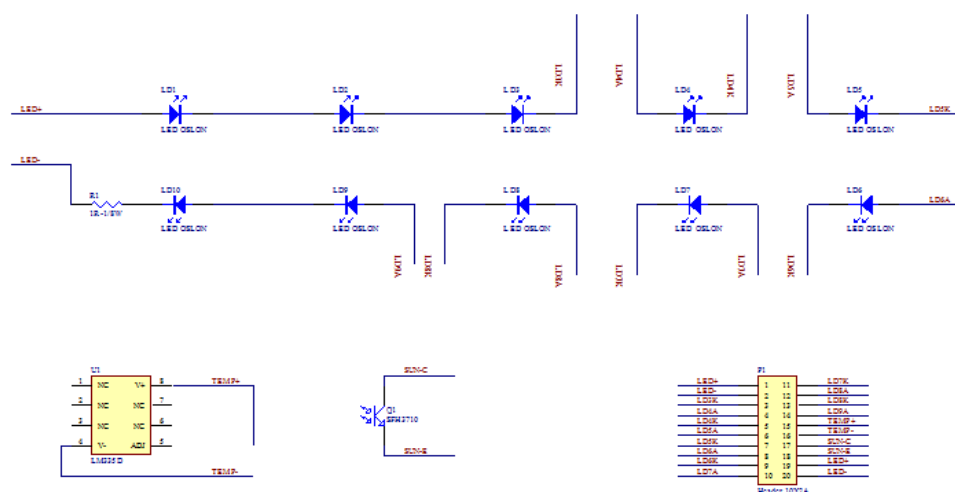


Figura 6.1 - Esquemático final da placa de LEDs do protótipo.

Neste circuito recorreu-se ao uso de componentes que são SMD (Surface Mount Device), que favorece no caso dos LEDs a dissipação térmica. O circuito impresso resultante é constituído por duas camadas, *top layer* e *bottom layer*, e está representado na Figura 6.2.

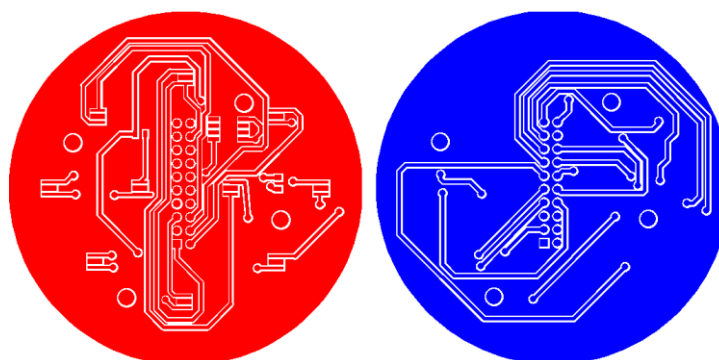


Figura 6.2 - Circuito impresso final da placa de LEDs do protótipo. (*Top layer* a vermelho e *bottom layer* a azul)

O desenho desta placa teve dois pormenores em conta, o primeiro a distribuição dos LEDs o outro facilitar a dissipação térmica dos LEDs. A distribuição dos LEDs é feita de forma circular para que o fluxo luminoso tenha uma saída da mesma forma. Isto porque a luminária proposta deverá ser utilizada para substituir luminárias convencionais que têm uma configuração circular. A dissipação dos LEDs é um aspecto fundamental para prolongar a vida útil da luminária. Para isso criou-se um plano de cobre tanto na parte superior como na inferior para que esse plano seja utilizado como condutor térmico para o dissipador onde esta placa será aplicada.

Placa de driver

O esquemático desenhado para a placa driver é apresentado na Figura 6.3.

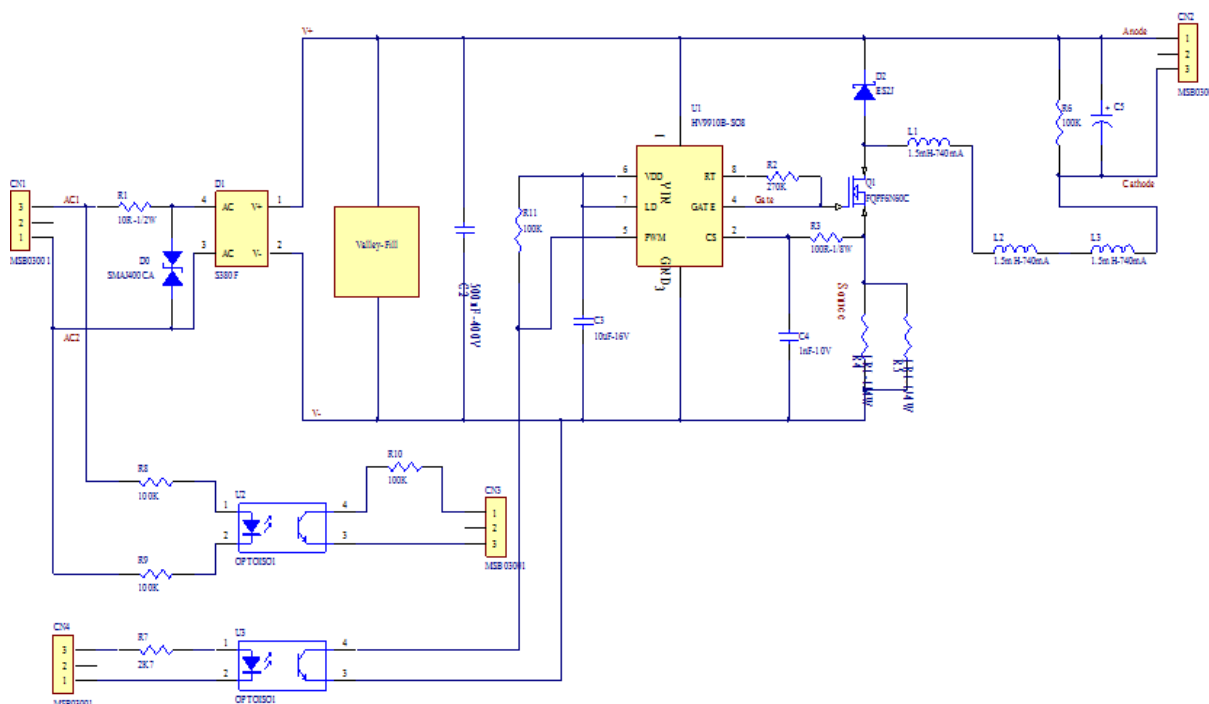


Figura 6.3 - Circuito final da placa de driver.

O circuito impresso resultante é constituído por duas camadas, *top layer* e *bottom layer*, e está representado na Figura 6.4.

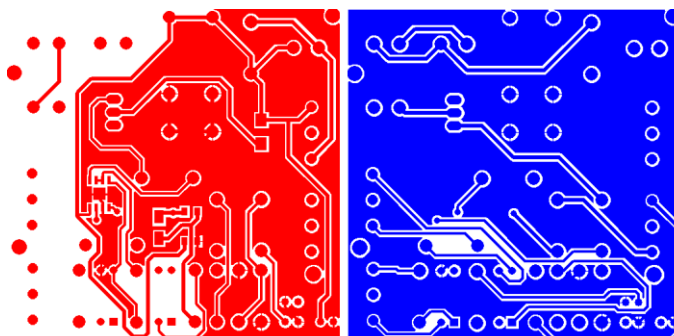


Figura 6.4 - Circuito impresso final da placa de driver do protótipo. (*Top layer* a vermelho e *bottom layer* a azul)

O desenho desta placa teve em consideração a zona de comutação. Esta zona é uma zona bastante crítica no que diz respeito a circuitos com frequências de comutação elevada. Os condensadores de entrada ficaram colocados neste ponto, para que os picos da comutação não se propaguem pela placa e sejam absorvidos logo na zona onde são gerados.

6.1.1. Protótipo com módulo de gestão e comunicação

A Figura 6.5 mostra o protótipo assembled com todos os seus módulos.

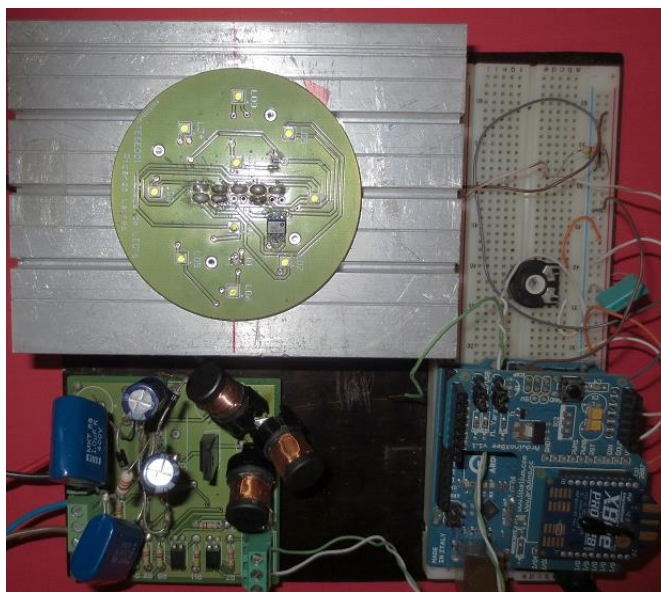


Figura 6.5 - Circuito impresso final da placa de driver do protótipo todo montado.

6.2. Teste eléctrico

Para avaliar o comportamento eléctrico do circuito dimensionado, vários testes e medições foram efectuados. A lista de equipamentos utilizados durante o teste é a seguinte,

- Power meter Yokogawa DT-210
- Osciloscópio Owon MS07102T
- Multímetro Kayse MY-64

6.2.1. Comportamento da placa de LEDs

O comportamento da placa de LEDs é avaliado através da temperatura que os LEDs atingem. A Tabela 6.1 mostra esses valores após equilíbrio térmico.

	Temperatura do ponto de solda [°C]	Temperatura da junção [°C]
LED 1	82,7	93,4
LED 2	92,1	102,8
LED 3	94,3	104,9
LED 4	81,4	92,1

	Temperatura do ponto de solda [°C]	Temperatura da junção [°C]
LED 5	82,6	93,3
LED 6	83,1	93,8
LED 7	85,3	95,9
LED 8	91,6	102,3
LED 9	83,8	94,5
LED 10	95,3	105,9

Tabela 6.1 - Temperaturas dos LEDs aplicados na placa com dissipador.

O diferencial de temperatura pode justificar pela junção da placa de LEDs e o dissipador. Em alguns pontos essa junção não é a ideal, o que faz com que haja LEDs mais quentes que os outros. Para avaliar a vida útil da luminária, a temperatura da junção é fundamental. Para isso tiramos a temperatura mais alta e assim fazemos o cálculo da vida útil, que é de aproximadamente 105 °C visto que a resistência térmica da junção é de 9,2 K/W.

Recorrendo à informação dada pelo fabricante do LED,

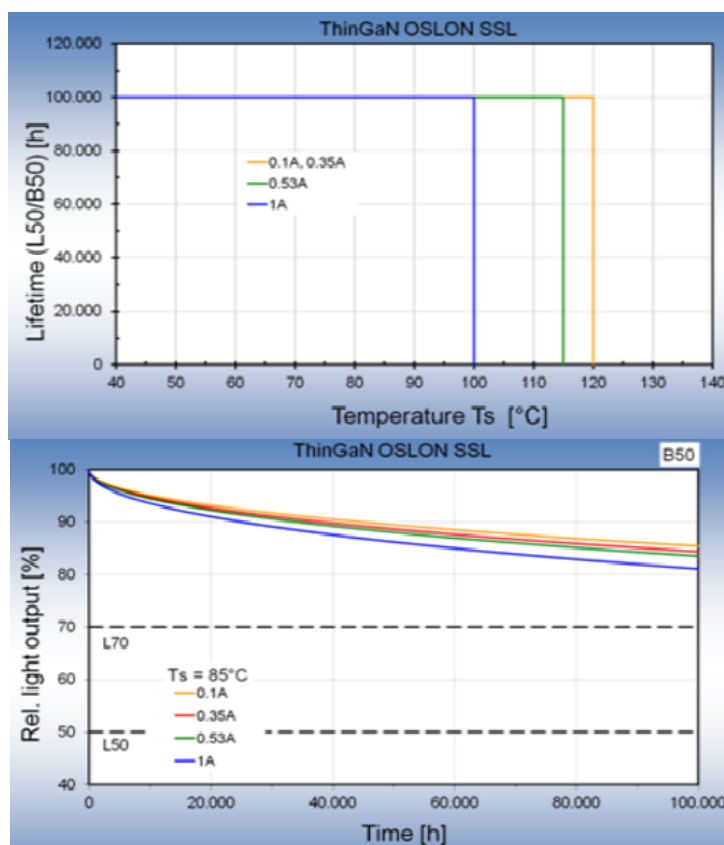


Figura 6.6 - Temperatura vs Horas de vida útil, [6].

Que a linha de referencia é a linha verde, o que se concluir que a vida útil da luminária é de 100.000 horas. A vida útil pode-se calcular pela duração prevista pelo fabricante do LED, ou diz-se que o LED atingiu a sua vida útil quando atinge os 70% do fluxo luminoso inicial. Quando o LED chega ao fim

da sua vida útil, não significa que o LED se apaga mas sim que já não fica garantido o fluxo luminoso ideal.

6.2.2. Comportamento do driver

Para avaliar o comportamento do driver, vários pontos têm de ser medidos. Estas medições foram efectuadas com a corrente de saída no seu valor máximo.

Deteção da corrente

Um dos pontos será o ponto de controlo de corrente, onde poderá ser visto o comportamento do oscilador. A Figura 6.7 mostra o resultado no pino CS no HV9910b.

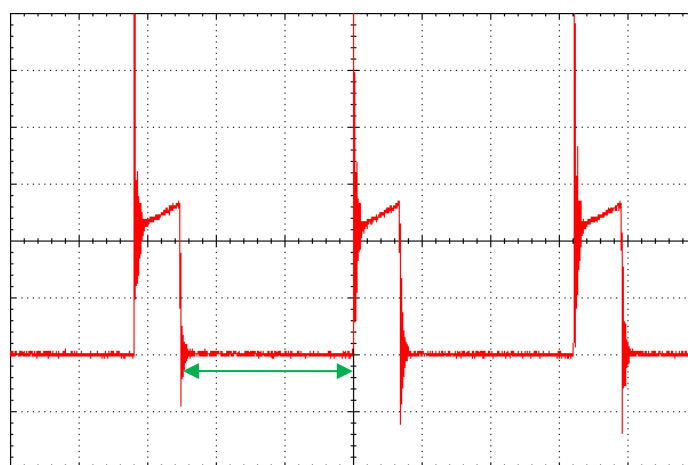


Figura 6.7 - Forma de onda do pino CS do HV9910b. (Escala: Tensão:100mV/Div. Tempo:5us/Div.)

Podemos observar que quando o transístor liga temos um pico provocado pelo arranque do mesmo. Em seguida temos a tensão a aumentar em rampa, ou seja representa a corrente que atravessa a resistência de *sense*. Quando chega cerca dos 270 mV o transístor desliga, isto significa que a corrente chega ao valor de pico definido. O fabricante anuncia um valor máximo de 275 mV de patamar máximo, e estando nos 270 mV tira-se a conclusão que o *driver* está a ter o funcionamento correcto.

T_{off}

Outro pormenor que se pode observar na imagem anterior e na Figura 6.8, que está representado a verde, é T_{off} fixo.

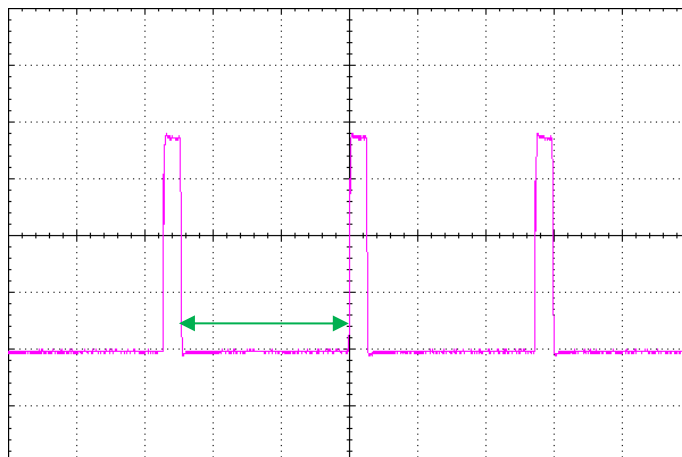


Figura 6.8 - Forma de onda da gate do transistor. (Escala: Tensão:2V/Div. Tempo:5us/Div.)

No dimensionamento foi definido um T_{off} de 9,92 μs , mas na realidade o T_{off} ficou aproximadamente nos 12,5 μs . Esta variação pode ser justificada por duas principais razões. A primeira é a tolerância de 5% da resistência de 270 k Ω . A segunda razão é a tolerância do condensador interno do circuito integrado. O fabricante não especifica qual a tolerância, mas por experiências anteriores conclui-se que existe uma tolerância elevada. Apesar da diferença o funcionamento não foi posto em causa, tendo como consequência o valor de *ripple* na corrente de saída como veremos de seguida.

Observamos também que a tensão à *gate* é de 7,5 V.

Tensão na bobine

A Figura 6.9 representa o andamento da tensão na bobine, tendo em conta que a medição foi feita com a ponta de prova a x10.

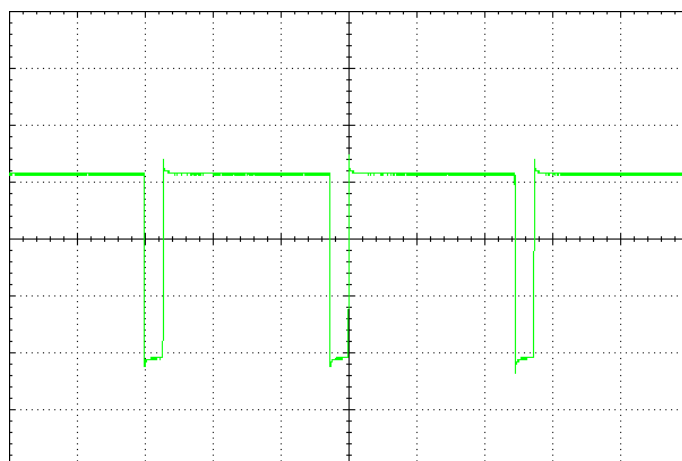


Figura 6.9 - Forma de onda da gate do transistor. (Escala: Tensão:100mV/Div. Tempo:5us/Div.)

Observamos que a tensão da bobine baixa significa que o transistor começa a conduzir, e quando o transistor está ao corte a tensão resultante na bobine fica aproximadamente dos 32 V que é para alimentar a série dos LEDs menos a queda de tensão do díodo ES2J, que tem uma queda de tensão a rondar os 1.1 V.

Corrente de saída

Para fazer a medição da corrente de saída, e da bobine, recorreu-se ao uso de uma resistência de $1\ \Omega$ com precisão de 1%. A figura 6.10 mostra como foi efectuada a medição.

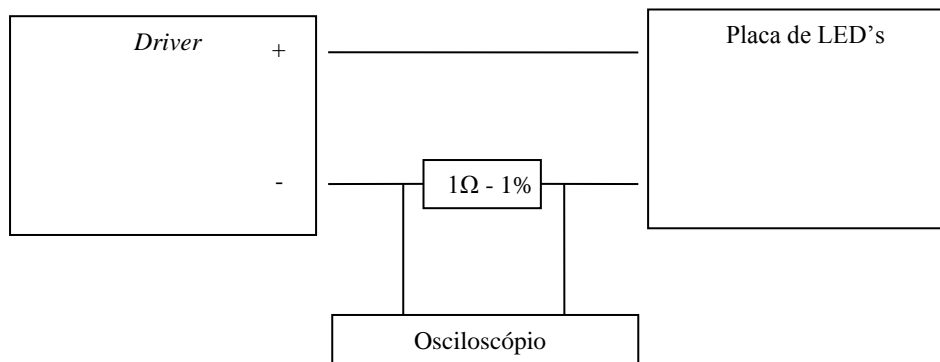


Figura 6.10 - Esquema de ligações para medição da corrente dos LEDs.

A Figura 6.11 representa a corrente que está a ser fornecida aos LEDs.

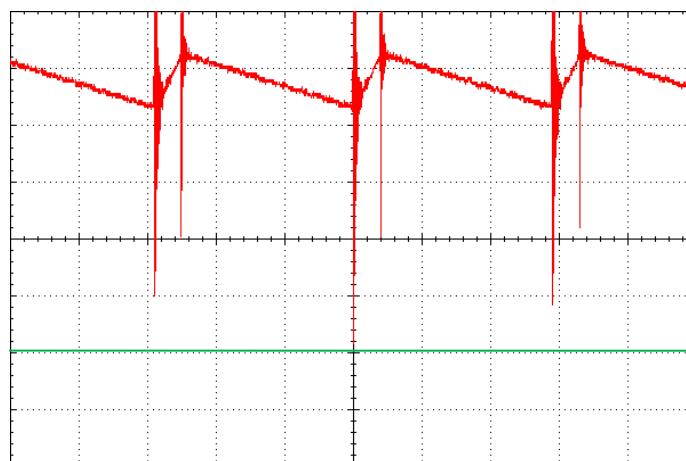


Figura 6.11 - Forma de onda da corrente fornecida aos LEDs. (Escala: Tensão:100mV/Div. Tempo:5us/Div.)

Com a origem representada pela linha verde, observamos que a forma de onda da corrente tem um valor máximo aproximadamente de 520 mA e um valor mínimo de 430 mA. Com isto temos um valor médio de 475 mA, com um *ripple* de 19%, ou seja 90 mA.

Os valores previstos no dimensionamento, era um valor médio de 470 mA com 15% de *ripple*. Com isto teríamos um valor máximo de 505,25 mA e um valor mínimo de 434,75 mA.

A diferença relativamente ao valor de pico justifica-se com o uso da resistência de *sense* maior que a original calculada. Como foi dito no capítulo do dimensionamento, usou-se uma resistência equivalente de $0,55\ \Omega$ ao invés de $0,5\ \Omega$. Isto faz com que o valor de pico atinja um valor superior ao esperado. Outro motivo, é o facto de o T_{off} ser superior ao teórico. Ou seja com isto o valor de máximo e o valor mínimo ficam alterados, alterando também o valor do *ripple*. Mas com isto atinge-se o valor

médio muito próximo ao dimensionado o que faz com que o circuito esteja a funcionar como esperado.

Comportamento da corrente relativamente ao PWM

Em seguida é observado o comportamento da corrente de saída com variação do controlo de PWM fornecido pelo micro controlador.

PWM a 100%

A corrente de saída com o PWM a 100% apresenta a seguinte forma de onda.

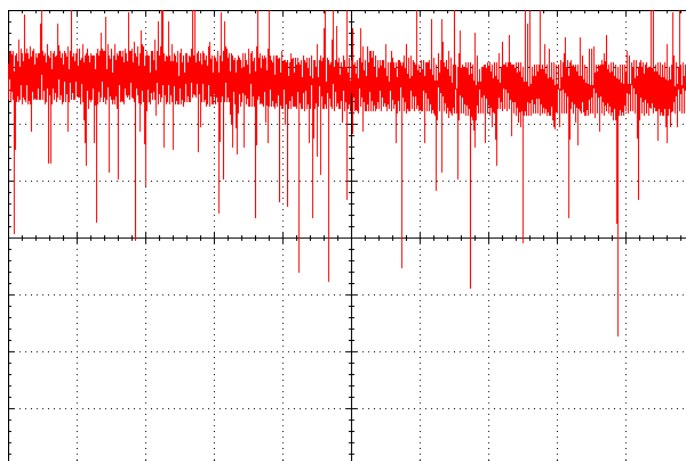


Figura 6.12 - Forma de onda da corrente com PWM a 100%. (Escala: Tensão:100mV/Div. Tempo:500us/Div.)

Ou seja o PWM a 100% faz com que a corrente tenha uma evolução constante ao longo do tempo, garantindo assim sempre os 475 mA de corrente média.

PWM a 75%

No caso de o PWM estar a 75% a corrente tem a evolução que está na figura 6.13.

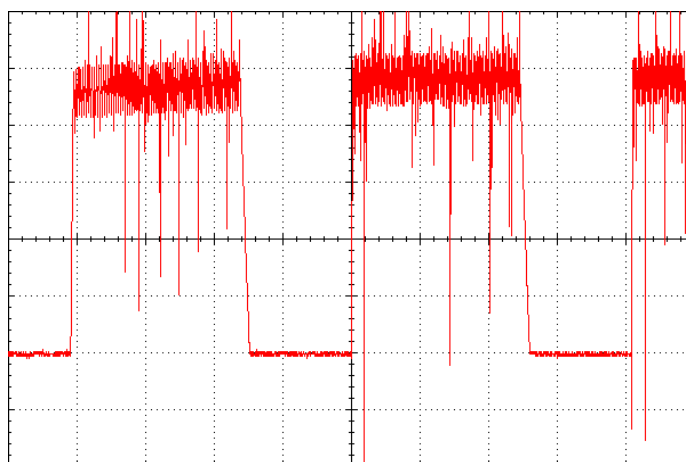


Figura 6.13 - Forma de onda da corrente com PWM a 75%. (Escala: Tensão:100mV/Div. Tempo:500us/Div.)

Podemos observar que a corrente só fica no pico durante algum período, em seguida o PWM fica a LOW e a bobine descarrega até chegar a 0. Isto traduz-se em um valor médio da corrente de 314 mA. O valor teórico previsto é de 356 mA o que representa um desvio de 11% com o valor real. Esta diferença justifica-se com o facto de o tempo de descarga da bobine ser consideravelmente rápido.

PWM a 50%

No caso de o PWM estar a 50% a corrente tem a evolução que está na figura 6.14.

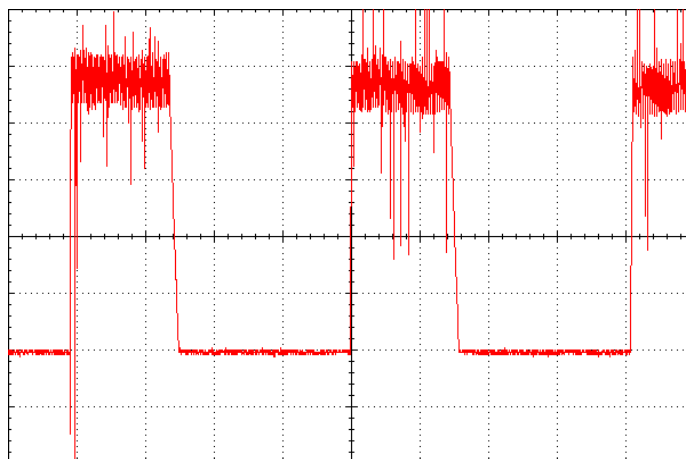


Figura 6.14 - Forma de onda da corrente com PWM a 50%. (Escala: Tensão:100mV/Div. Tempo:500us/Div.)

Podemos observar que a corrente só fica no pico durante algum período, em seguida o PWM fica a LOW e a bobine descarrega até chegar a 0. Isto traduz-se em um valor médio da corrente de 208 mA. O valor teórico previsto é de 238 mA, o que representa um desvio de 12,6% com o valor de real.

Resultados da eficiência do driver

A Tabela 6.2 mostra os resultados obtidos durante os testes. Estes testes foram feitos com o fluxo luminoso da luminária a 100%.

Tensão de entrada [V@50Hz]	Potência de entrada [W]	Factor de potência	Potência de saída [W]	Eficiência [%]
190	15,50	0,95	14,396	92,877
200	15,59	0,952	14,396	92,341
210	15,68	0,951	14,396	91,811
220	15,79	0,95	14,396	91,172
230	15,89	0,949	14,396	90,598
240	15,98	0,946	14,396	90,088
250	16,08	0,945	14,396	89,527
260	16,19	0,94	14,396	88,919

Tabela 6.2 - Resultados dos testes efectuados ao driver a 100%.

Pela Tabela 6.2 observamos que a potência de saída não se altera com a tensão de entrada. O fabricante anuncia um desvio da corrente de saída de 0,1%, mas não foi perceptível nas medições. Por sua vez a potência de entrada altera-se com a tensão de entrada, alterando-se também o factor de potência. Isto é devido ao aumento das perdas no circuito do driver.

Em termos da eficiência do *driver*, podemos observar que temos uma eficiência a tensão nominal de 90,6%. Este valor obtido é bastante relevante, tendo em conta a simplicidade do *driver*. Este valor foi atingido também recorrendo ao uso dos componentes indicados e com características que permitem que no circuito as perdas sejam reduzidas o mais possível.

Correcção do factor de potência

O funcionamento do *driver* utilizado o factor de potência ronda os 0,5. Para otimizar o factor de potência foi implementado o filtro Valley-Fill, obtendo-se um factor de potência de 0,949 o que indica que da luminária pode ser instalada em edifícios em que o requisito mínimo seja de 0,8.

6.3. Teste fotométrico

Para avaliar o comportamento fotométrico da luminária proposta foi feito o teste em um goniofotómetro modelo LMT GO-V 1900. Com o auxílio do software Limes 2000 extraíram-se as figuras que observaremos e avaliaremos.

Este teste foi realizado com uma inter-distância entre medições (ângulos *gama*) de 5°, e uma inter-distância entre planos (*C-Planes*) de 15°. Estas inter-distâncias permitem determinar com exactidão o comportamento fotométrico da luminária.

Este teste fotométrico gerou um ficheiro com extensão .ldt que permite que a informação extraída durante o teste seja utilizado em softwares de iluminação. Este ficheiro contém diagramas de radiação, comportamento eléctrico da luminária e constrói a tabela de intensidades luminosas ao longo dos vários planos. Com esta informação teremos tudo sobre a luminária a nível fotométrico.

6.3.1. Diagramas de radiação

Para visualizar a quantidade de luz que uma luminária consegue emitir pode ser observada através de dois diagramas, o primeiro o diagrama polar e o outro o diagrama cartesiano. Ambos traduzem a mesma informação mudando só a facilidade de leitura dos parâmetros.

O diagrama polar e cartesiano de radiação resultante do protótipo da luminária é o seguinte,

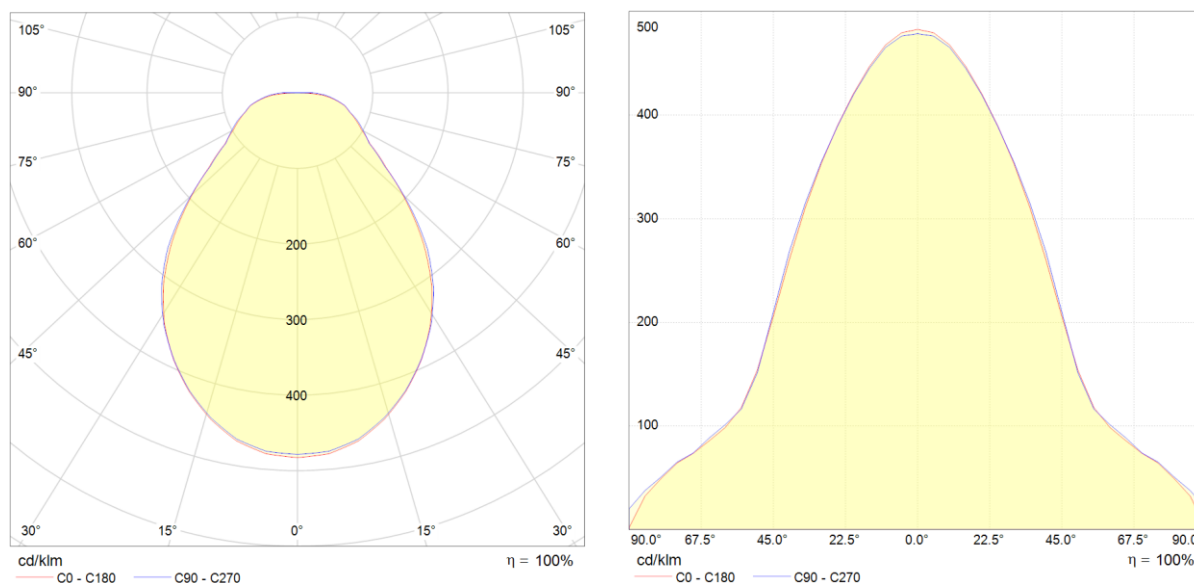


Figura 6.15 - Diagrama polar e cartesiano fotométrico do protótipo da luminária proposta.

Podemos observar que nos eixos existe uma simetria. Isto é resultado da distribuição de LEDs utilizada na placa em forma circular que garante uma uniformidade ao longo dos 360°. Podemos observar que no centro temos a intensidade luminosa concentrada, com intensidade cerca de 480 cd/klm, e que ao longo do ângulo de abertura isso vai-se dissipando. Isto é devido à abertura utilizada no LED. Como foi utilizado um LED com 80° de abertura, faz com que o LED seja direccional para onde esteja apontado.

6.3.2. Eficiência luminosa da luminária

O fluxo luminoso máximo que a luminária de protótipo atingiu foi de 1289 lm, sendo o fluxo esperado através do dimensionamento de 1400 lm. Esta diferença pode-se justificar com o facto do *top overlay* da placa de LEDs ser verde. Em testes realizados anteriormente a outros protótipos apresentaram um diferencial, relativamente à cor verde e branca do *top overlay*, de aproximadamente 5% relativamente ao fluxo luminoso. Isto pode representar um aumento de 64lm, que daria para atingir os 1353lm o que seria mais aproximado ao esperado no dimensionamento.

Para o fluxo luminoso que a luminária atingiu, consumindo 15,89 W para a tensão de alimentação nominal de 230 V, a eficiência luminosa atingida é de 81,12 lm/W. Esta eficiência é bastante considerável, comparativamente com as luminárias disponíveis no mercado. Outro facto de realçar, é que a eficiência atingida é conseguida com a aplicação de um *driver* simples o que demonstra a eficiência energética dos sistemas de iluminação a LED.

6.4. Avaliação energética de um sistema de luminárias

Para efectuar a avaliação energética de um sistema de luminárias, tendo em conta um fluxo luminoso desejado, toma-se como exemplo de uma sala com 99 m² (comprimento:11m e largura:9m). A altura da sala foi definida a 3 m.

A área a iluminar representa só uma parte da área da sala, ou seja 35 m^2 , e a iluminância a $0,8\text{m}$ do solo, ou seja a área de trabalho, terá de ser pelo menos 500 lx ao longo da área a iluminar. Este valor é o aconselhado para áreas de trabalho como por exemplo escritórios.

Para efectuar estes cálculos, foi utilizado o software de iluminação Dialux. Este software permite, com o ficheiro “.ldt” com as características da luminária, visualizar os rendimentos das luminárias e fazer cálculos luminotécnicos de áreas a iluminar.

6.4.1. Sem sistema de gestão aplicado

Para realizar esta avaliação será feita a comparação entre a luminária protótipo e uma luminária convencional com lâmpada CFL. A luminária convencional escolhida foi do fabricante Indelague com o modelo DYNA 9022F226A. Esta luminária tem uma potência total de 54 W , uma T_c de 4000 K e um CRI de 80.

Sistema com iluminação convencional

Com o auxílio do Dialux a distribuição das luminárias convencionais e as linhas isográficas resultantes da distribuição das luminárias ficou a seguinte,

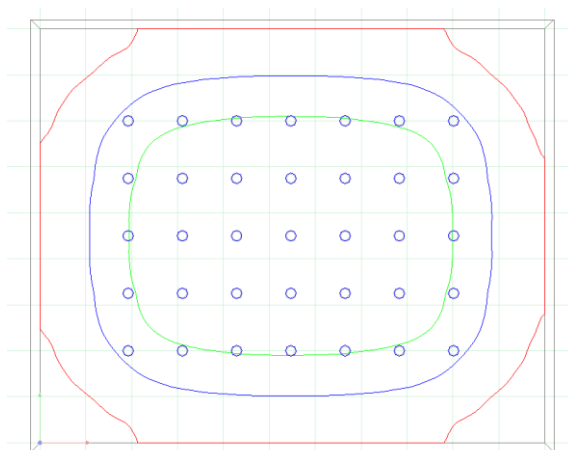


Figura 6.16 - Distribuição das luminárias convencionais e representação das linhas isográficas de iluminância.

Para garantir na área a iluminar uma iluminância de 500 lx , será necessário utilizar 35 luminárias. A potência instalada na instalação tem um total de 1890 W .

Assumindo que o ciclo de utilização é de 8 horas obtemos que a instalação consumirá $15,12 \text{ kWh}$.

As linhas isográficas representam a área interior com esse nível de iluminância. A verde está representada a linha dos 750 lx , a azul está representada a área dos 500 lx e a vermelho está representado a área dos 250 lx . Podemos observar que a área a iluminar está toda com pelo menos 500 lx .

Sistema com luminária protótipo a LED

Utilizando o mesmo método do ponto anterior, mas usando luminárias protótipo a LED a distribuição das luminárias e as linhas isográficas resultantes são as seguintes,

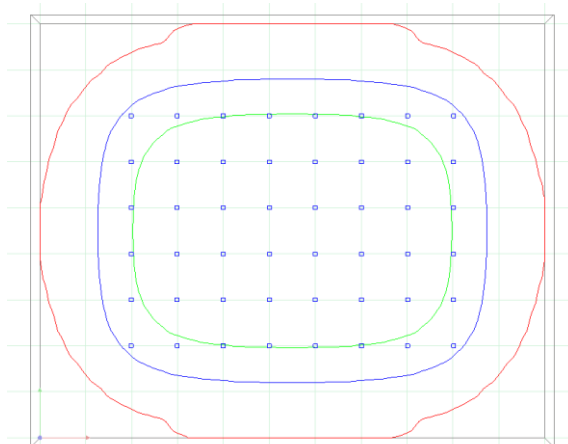


Figura 6.17 - Distribuição das luminárias protótipo a LED e representação das linhas isográficas de iluminância.

Com a utilização das luminárias protótipo é necessário utilizar 46 luminárias. O número de luminárias a utilizar é maior quando comparado com a luminária convencional, mas a potência instalada na instalação tem um total de 768 W, o que representa somente 40,6% da potência instalada com a luminária convencional demonstrada no ponto anterior, o que representa uma poupança de 59,4%. Assumindo que o ciclo de utilização é de 8 horas obtemos que a instalação consumirá 6,14 kWh. Podemos observar que a área a iluminar está toda com pelo menos 500lx, logo ficam atingidos os pressupostos definidos.

A figura seguinte representa a luminárias com o seu fluxo luminoso a 100% nos seus locais de aplicação, permitindo visualizar o resultado final do espaço com todas as luminárias. Na figura 6.18 estão também representadas as curvas fotométricas de cada luminária.

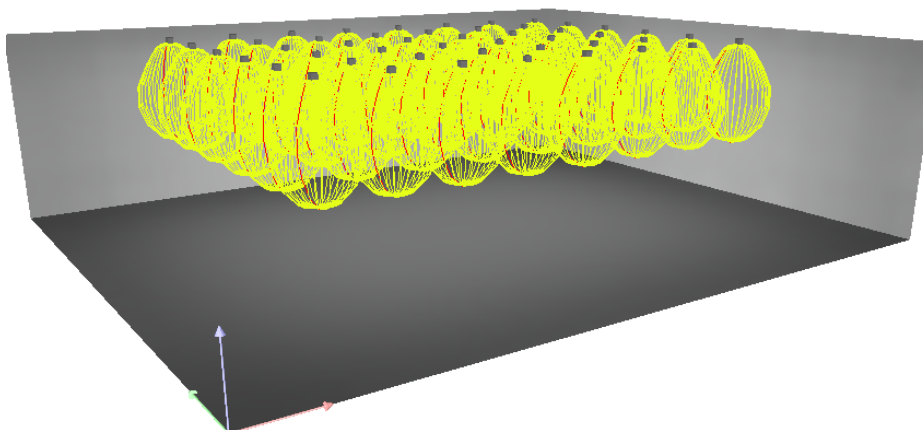


Figura 6.18 - Distribuição das luminárias protótipo a LED e representação das linhas isográficas de iluminância.

6.4.2. Sistema de gestão e comunicação aplicado

Com o sistema de gestão e de comunicação implementados, será feita a mesma análise que foi efectuada nos pontos anteriores. A figura 6.19 mostra um cenário possível de utilização do espaço.

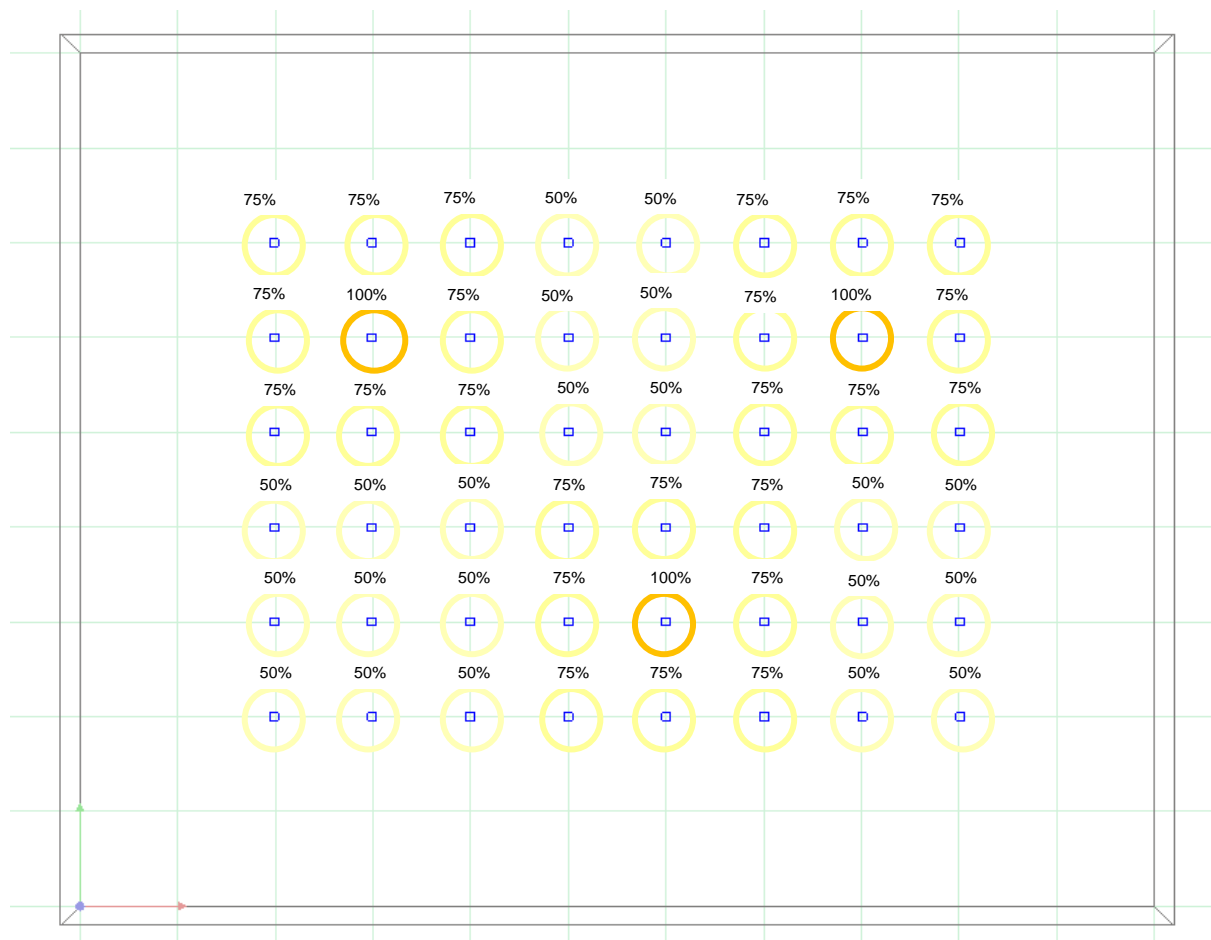


Figura 6.19 - Cenário de utilização da sala com luminárias protótipo e sistema de gestão e comunicação implementados.

Como foi descrito no dimensionamento, as luminárias têm um comportamento de quando têm o sensor de presença activo o seu fluxo luminoso fica a 100%, as circundantes ficam a 75% e as restantes ficam a 50%. No cenário proposto temos três pessoas na sala, logo vamos ter três luminárias com o seu fluxo a 100%. Teremos 24 luminárias a 75% e 21 luminárias somente a 50%. Falando em termos de potências em causa, temos que a 100% a luminária consome 15,89 W, a 75% consome 10,02 W e a 50% consome 6,2 W.

Para o consumo do módulo de gestão com o módulo de comunicação, assumiu-se 500 mW. Isto porque o módulo de comunicação tem uma potência máxima de 132 mW, um sensor *pir* (*passive infra-red*) pode ter um consumo médio de 50 mW e o restante vamos assumir que é consumido por o micro controlador.

Na Tabela 6.3 está representado as potências consumidas no cenário proposto.

Fluxo luminoso [%]	nº de lumárias	Potência consumida s/gestão [W]	Potência consumida c/gestão [W]
100	3	47,67	49,17
75	24	240,48	252,48
50	21	130,2	140,7
	Potência total	418,35	442,35

Tabela 6.3 - Valores resultantes do cenário proposto com a luminária protótipo completa.

Pela tabela obtemos que a instalação consumirá cerca de 418W se não tivermos em conta com o consumo do sistema de gestão. Se considerarmos o sistema de gestão temos que a instalação consome cerca de 442. Podemos tirar uma conclusão que o sistema de gestão tem um consumo bastante reduzido, representa cerca de 5,4% da instalação, quando comparamos com as vantagens em termos de eficiência energética que conseguimos atingir.

Assumindo que o ciclo de utilização é de 8 horas obtemos que a instalação consumirá 3,54 kW/h. Em termos de quantidade, garante-se que a área ocupada respeita os pressupostos iniciais visto que a luminária por cima da área está com o fluxo luminoso no máximo e é acompanhada com as luminárias circundantes a 75%.

6.5. Análise da avaliação energética dos sistemas de luminárias

Se comprarmos os dois primeiros casos, vemos que conseguimos uma poupança de 59,4% o que é bastante significativos e logo à partida conseguimos concluir que o uso de luminárias a LED traz uma enorme vantagem relativamente à iluminação convencional. Apesar do investimento inicial mais elevado, visto que as luminárias a LED tem um custo superior à iluminação convencional, ao longo de um custo espaço de tempo o investimento vai ser absorvido pelo que se consegue economizar em energia eléctrica. Outro ponto que temos de considerar, é a vida útil das luminárias que no caso do LED é bastante superior à iluminação convencional. Isto aliado à poupança energética obtemos poupanças económicas bastante acentuadas.

Ao aplicar o sistema de gestão a poupança entre o sistema de iluminação a LED sem gestão e com gestão representa uma poupança de 42,4% para o caso de aplicação dado. Se considerarmos espaços com maiores dimensões e com uma utilização irregular este valor pode chegar a valores superiores.

Se compararmos o caso de luminárias a LED com gestão e a iluminação convencional, para o caso dado, obtemos uma poupança de 77,6%.

Notoriamente que é uma opção a considerar o uso da iluminação a LED com gestão, tanto em novas instalações ou substituição da iluminação convencional, visto conseguimos poupanças energéticas acima dos 77%.

7. Conclusão e trabalhos futuros

7.1. Conclusões

Conforme referenciado no início da Dissertação, a iluminação pública em espaços interiores representa um considerável consumo energético agravado pela elevada ineficiência energética que caracteriza os sistemas de iluminação convencional. Neste contexto, foi proposto uma solução tecnológica de iluminação inteligente que visa melhorar significativamente o índice de eficiência energética, assente em dois vectores de desenvolvimento:

- utilização do LED como fonte de luz e respectiva optimização dos circuitos electrónicos de alimentação e controlo, e
- integrar os meios de comunicação rádio necessários ao estabelecimento de uma rede colaborativa entre luminárias, por forma a ajustar e controlar dinamicamente os níveis de luminosidade.

No protótipo apresentado, demonstrou-se, com sucesso, a operacionalidade de um sistema de iluminação LED que pode funcionar com ou sem gestão e comunicação integrada. Foi igualmente mostrado que com a solução proposta é possível um aumento da vida útil podendo atingir 100.000 horas, garantindo uma redução de custos de manutenção do sistema de iluminação. Os testes realizados indicam que a luminária apresenta um factor de potência de 0,949, o que garante a utilização deste sistema em qualquer tipo de instalação pretendida.

Com base nos consumos individuais de cada luminária, uma redução do consumo, face a luminárias convencionais, que pode ascender aos 59%. Todavia, esta redução pode atingir os 77% se for ativado modo de controlo distribuído entre luminárias. Isto demonstra que a utilização de uma luminária LED em modo on-off *standalone* permite poupanças bastante significativas. Por outro lado, conclui-se que ao adicionar um simples módulo de comunicações rádio, permitindo a integração de um algoritmo de gestão inteligente, é ainda possível um poupança extra dos 42%, quando comparado com o sistema LED simples.

Ao aplicar o sistema de gestão e de comunicação, a poupança entre o sistema de iluminação LED sem gestão e com gestão representa uma poupança de 42,4% para o caso em estudo. Se considerarmos espaços com maiores dimensões e com utilização irregular, este valor pode ser ainda superior, justificando a utilização de um mecanismo de gestão integrado. Por último, cm este sistema é possível atingir melhor qualidade e maior conforto da iluminação, visto que o sistema uniformiza a zona ocupada garantindo o nível luminoso desejado.

7.2. Trabalhos futuros

Na luminária

Ao nível do *driver* poderá ser explorada e optimizada a topologia do tipo *flyback*, por forma a melhorar o rendimento do circuito de alimentação. Explorar novas configurações da disposição dos

leds na luminária. Por último, explorar formas de conversão da energia térmica novamente em energia eléctrica.

No sistema de gestão

Ao nível do sistema de gestão, poderá ser reforçada a interacção entre luminárias, nomeadamente na caracterização conjunta do nível de luminosidade do espaço interior. Uma segunda alteração passaria pelo controlo mais fino do fluxo luminoso deixando de ter apenas três fases evitando, por exemplo, que o fluxo luminoso fique a 100% quando não é necessário. Esta funcionalidade poderá aumentar significativamente a eficiência energética.

No sistema de comunicação

Ao nível do sistema de comunicação, sugere-se o desenvolvimento de um mecanismo do tipo “*plug-and-play*”, por forma a facilitar todo o processo de instalação e configuração. Ou seja, quando um novo dispositivo fosse instalado na rede a detecção da rede seria automática e não forçada por programação como feito na primeira versão. Isto poderia ser implementado com base numa chave fixa para autenticação e validação, e assim o coordenador validaria esta chave, permitindo o registo de um novo dispositivo na rede.

Outros dos aspectos a explorar é o mecanismo de *broadcast* permitindo o mapeamento dinâmico sobre o estados de todas as luminárias. Por último, a integração num sistema de segurança através da rede permitiria uma utilização indirecta da luminária como elemento de detecção de intrusão.

Bibliografia

- [1] – EDP (2012). “*Origens da energia*”. Disponível: <http://www.edpsu.pt/pt/origemdaenergia/Pages/OrigensdaEnergia.aspx>. Acedido a: 15-09-2013.
- [2] – Empresa X. (2012) “Estudo das fontes de consumo energético”. Lisboa. Empresa X.
- [3] – Empresa X. (2012). “Estudo sobre conversão para luminárias LED”. Lisboa. Empresa X.
- [4] – Aulas teóricas (2011). “*Teoria básica das ondas electromagnéticas*”. Disponível: <http://paginas.fe.up.pt/~ee86027/Teoria%20b%C3%A1sica%20das%20ondas%20eletromagn%C3%A9ticas.pdf>. Acedido a 23/07/2013.
- [5] – Marco Aurélio da Silva. (?). “*O que são ondas electromagnéticas*”. Disponível: <http://www.brasilecola.com/fisica/o-que-sao-ondas-eletromagneticas.htm>. Acedido a: 25-05-2013.
- [6] - OSRAM. (2012). “*Iluminação: Conceitos e Projectos - Parte 1*”. Disponível: http://www.osram.com.br/osram_br/Ferramentas_&_Catlogos/_pdf/Arquivos/Iluminacao_Geral/Manual_do_Curso_Iluminacao,_Conceitos_e_Projetos/Manual_Luminotecnico_-_parte_01.pdf. Acedido a: 23/05/2013.
- [7] – Osram (2011) “*Manual luminotécnico prático*”. Worldwide: Osram.
- [8] – (OSRAM 2013) – Disponível: <http://catalog.osram-os.com/catalogue/catalogue.do?act=showBookmark&favOid=0000000700039de3029b003a>. Acedido a: 10/08/2013.
- [9] - OSRAM (2012). “*Iluminação: Conceitos e Projectos – Parte 2*”. Disponível: http://www.osram.com.br/osram_br/Ferramentas_&_Catlogos/_pdf/Arquivos/Iluminacao_Geral/Manual_do_Curso_Iluminacao,_Conceitos_e_Projetos/Manual_Luminotecnico_-_parte_02.pdf. Acedido a: 23/05/2013.
- [10]- Dick Erdmann. “*Color Rendering Index*”. Worldwide. General Electric.
- [11] - Commission Internationale de l’Eclairage. “Method of Measuring and Specifying Colour Rendering Properties of Light Sources”. CIE. 13.3-1995.
- [12] – Nist. (2010). “*Color rendering of light sources.*” Available: <http://physics.nist.gov/Divisions/Div844/facilities/vision/color.html>. Acedido a: 26/06/2013.
- [13] - Osram (2011). *LED Fundamentals - Radiometry and Photometry*. 2nd ed. Osram.
- [14] - Schreder (2013). “*Conceitos básicos de iluminação*”. Disponível: <http://www.schreder.com/pts-pt/LearningCenter/ConceitosBasicosDeIluminacao/Pages/default.aspx>. Acedido a: 22/07/2013.
- [15] – IES (1995), Illuminating Engineering Society of North America. 8th edition. New York. IESNA.
- [16] – Osram (2013). “*Catalog*”. Disponível: <http://catalog.myosram.com/>. Acedido a: 22/07/2013.
- [17] - Silva, Mauri Luíz (2002). “*Luz Lâmpadas & Iluminação*”. 3rd ed. Rio de Janeiro: Ciência Moderna.
- [18] – Zukauskas, A., Shur, M., Gaska, M. (2002). “*Introduction to solid-state lighting*”. New York: John Wiley & Sons, Inc.
- [19] – Schubert, E. Fred (2006). “*Light-Emitting Diodes*”. 2nd ed. New York: Cambridge University Press.

- [20] – Sedra, Adel S., Smith, Kenneth C. (2004). “*Microelectronic Circuits*”. 5th ed. New york: Oxford University Press.
- [21] - ETAP. (2013). “*Maintenance factor for LED lighting*”. Available: www.etaplighting.com. Last accessed 04/08/2013.
- [22] - Osram. (2012). “*Reliability of the OSLOM SSL and OSLOM Square Product Groups*”. Disponível: www.osram.com. Acedido a:04/08/2013.
- [23] - Osram. (2003). *5mm LED, Diffused*. Disponível:<http://images.ihcontent.net/vipimages/VipMasterIC/IC/OSOS/OSOSS00730/OSOSS00730-1.pdf>. Acedido a:10/08/2013.
- [24] – Fairchild. (2013). *LM317*. Disponível: <http://www.fairchildsemi.com/ds/LM/LM317.pdf>. Acedido a:10/08/2013.
- [25] - Infineon. (2009). *BCR401R*. Disponível:<http://www.infineon.com/dgdl/bcr401r.pdf?folderId=db3a30431400ef68011407a9cfc70181&fileId=db3a30431400ef68011407ab7ea10185>. Acedido a:10/08/2013.
- [26] – Sueker, Keith H. (2005). “*Power Electronics Design - A Practitioner's Guide*”. Raleigh, NC: SciTech Publishing.
- [27] – Brown, Marty (1990). “*Practical Switching Power Supply Design*”. San diego, California: Academic Press.
- [28] – Rashid, Muhammad H. (2011). “*Power Electronics Handbook*”. 3rd ed. Burlington, MA: Elsevier, Inc.
- [29] – Linear Technology. (2012). “*LT3590*”. Disponível: <http://cds.linear.com/docs/en/datasheet/3590f.pdf>. Acedido a:12/08/2013.
- [30] – On Semiconductor (2002). “*Switchmode Power Supplies*”. 3rd ed. USA: On semiconductor.
- [31] - Hesener, Alfred. (2011). “*Electromagnetic Interference in Power Supplies*”. Disponível: <http://www.fairchildsemi.com/Assets/zSystem/documents/collateral/onlineSeminars/Electromagnetic-Interference-EMI-in-Power-Supplies-WP.pdf>. Acedido a:13/08/2013.
- [32] - Berman, Mel. (2008). *All about EMI filters*. Disponível: <http://www.us.tdk-lambda.com/lp>. Acedido a:13/08/2013.
- [33] - Grigore, Vlad (2001). *Topological Issues In Single-Phase Power Factor Correction*. Helsinki: Institute of Intelligent Power Electronics.
- [34] – Nunes, Renato, “Edifícios Inteligentes: Conceitos e Serviços”. DEEC,IST/INESC.
- [35] - Halonen, Liisa (2010). “*Guidebook On Energy Efficient Electric Lighting for Buildings2*”. Finland: Aalto University School of Science and Technology.
- [36] - Lun-Wu Yeh, Che-Yen Lu, Chi-Wai Kou, Yu-Chee Tseng, and Chih-Wei Yi. (2010). “*Autonomous Light Control by Wireless Sensor and Actuator Networks*”. Disponível:<http://people.cs.nctu.edu.tw/~yctsen/papers.pub/sensor34-lamp-ieee-sensors.pdf>. Acedido a:20/08/2013.
- [37] - Singhvi, Vipul. “*Intelligent Light Control using Sensor Networks*”, *SenSys'05*, 2005:
- [38] USITT - United States Institute for Theatre Technology (2000), “DMX512-A - Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories”, Revision 3.
- [39] – Elation Professional (2008). “*DMX 512 Handbook*”. Worldwide: Elation Professional.

- [40] - Maxim Integrated. (2001). “RS-485 (EIA/TIA-485) Differential Data Transmission System Basics”. Disponível: <http://pdfserv.maximintegrated.com/en/an/AN736.pdf>. Acedido a: 14/08/2013.
- [41] - Lumikit. (2007). *DMX - Funcionamento básico*. Disponível: http://www.dmx512.com.br/02_funcionamento.htm. Acedido a: 14/08/2013.
- [42] - Nývlt, Ondřej (2011). “Buses, Protocols and Systems for Home and Building Automation”. Prague: Czech Technical University in Prague.
- [43] - Tridonic (2011). “DALI Manual”. 2nd ed. Austria: Tridonic.
- [44] - *DALI Specs Guide*. Disponível: http://ritelites.com/yahoo_site_admin/assets/docs/dali-specs-guide1.23264229.pdf. Acedido a: 15/08/2013.
- [45] - ZVEI, Division Luminares (2001). “DALI Manual”. Frankfurt: DALI AG.
- [46] - Artistic Licence Engineering Ltd . “The DALI Guide”. 2nd ed. Worlwide: Artistic Licence Engineering Ltd.
- [47] – Gislason, Drew (2008). “ZigBee Wireless Networking”. Wordwide: Newnes.
- [48] – Labiod, Houda (2007). “Wi-Fi, Bluethooth, Zig-Bee and WiMax”. Dordrecht, The Netherlands: Springer.
- [49] – Faludi, Robert (2010). “Building Wireless Sensor Networks with ZigBee, XBee, Arduino, and Processing”. North Sebastopol, CA: O'Reilly Media.
- [50] – Digi International Inc. (2009). “XBee®/XBee-PRO® ZB RF Modules”. Bren Road East Minnetonka, MN: Digi International Inc.
- [51] - Osram. (2013). “Oslon SSL 80”. Disponível: <http://catalog.osram-os.com/catalogue/catalogue.do;jsessionid=D44F2D8F3B3BA43134AECB4757A974E9?favOid=000000060001c4e3013c0023&act=showBookmark>. Acedido a: 26/07/2013.
- [52] - Osram. (2007). “Silicon NPN Phototransistor with V_{λ} Characteristics”. Disponível: <http://catalog.osram-os.com/catalogue/catalogue.do?act=showBookmark&favOid=00000001000010480002003a>. Acedido a: 26/07/2013.
- [53] - Texas Instruments. (1999). “LM335”. Disponível: <http://www.ti.com/lit/ds/symlink/lm135.pdf>. Acedido a: 26/07/2013.
- [54] –Supertex Inc.. (2010). “HV9910b”. Disponível: <http://www.supertex.com/pdf/datasheets/HV9910B.pdf>. Acedido a: 26/07/2013.
- [55] - Supertex Inc.. (2010). “AN-H48”. Disponível: www.supertex.com. Acedido a: 26/07/2013.
- [56] - Supertex Inc.. (2010). “AN-H50”. Disponível: www.supertex.com. Acedido a: 26/07/2013.
- [57] - Supertex Inc.. (2010). “AN-H64”. Disponível: www.supertex.com. Acedido a: 26/07/2013.

Anexo A - Zig-Bee Stack

O modelo usado numa comunicação Zig-Bee está representado na figura A1.1.

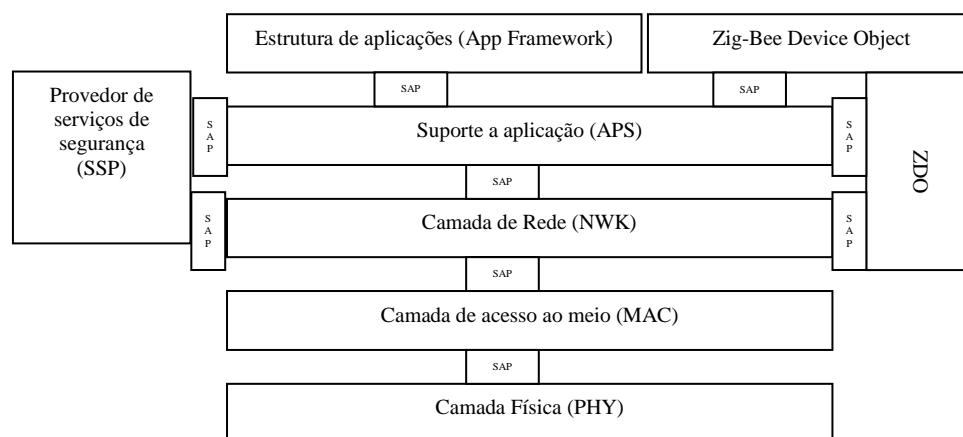


Figura A1.1 - Zig-Bee Stack.

As duas camadas inferiores são definidas segundo o protocolo 802.15.4, [47]. A camada física (PHY) tem como função activar/desactivar o hardware RF, garantir que a ligação tem uma boa relação força/qualidade de sinal, fazer a escolha do canal de comunicação.

A camada NWK tem como função gerar e criar a topologia da rede, gerir o dispositivo que entre e saia da rede, gerir o *routing* de pacotes, fazer o endereçamento na rede.

A camada de APS tem como função filtrar os pacotes de terminais não registados ou perfis que não correspondem à ZCL, manter a tabela de ligações tanto com *end-devices* como grupos definidos, manter a tabela de *routing*.

A estrutura de aplicações (App Framework) contém a Zig-Bee *Cluster Library* (ZCL), se necessário, e também faz as funções para as quais foi programado. Ou seja, é nesta camada que são realizadas as funções às quais os nossos *End-devices* foram destacados. A ZCL pode ser usada ou não consoante o que pretendemos da aplicação, e esta define um conjunto de comandos que podem ser suportados em múltiplos perfis. Por exemplo comandos como saber a temperatura de operação, tensão de operação, versões de firmware ou de hardware estão contemplados na ZCL. O uso da ZCL pode significar estar dentro das especificações da Zig-Bee *Alliance*, podendo actuar nas áreas certificadas pela entidade, como por exemplo, automação de edifícios. Apresenta contudo uma grande desvantagem, quando falamos no uso de micro controladores de 8 bits, visto que tem 3,8 Kb para um conjunto básico de operações. É com esta pilha de comandos que podemos conseguir utilizar o Zig-Bee *Light Link*, mas não foi considerado no âmbito da tese em questão.

Entre as camadas existe uma camada que faz o interface entre a camada superior e a inferior. Na realidade são duas camadas intermédias, uma para dados e outra para gestão, [47]. Estas camadas fazem com que o que se realiza na camada superior não interfira com a camada inferior.

A1.1. Estrutura e funções de cada dispositivo de uma rede Zig-Bee

Para formar uma rede Zig-Bee, existem três tipos de nós. Temos de ter pelo menos um coordenador da rede (ZC), temos um ou vários *routers* (ZR) e temos vários *End-Devices* (ZED), [47].

Funcionalidades do Coordenador (ZC)

O módulo ZC tem várias funcionalidades como formar a rede, escolha do canal de comunicação, escolha do PAN ID, gerir e definir a segurança da rede, gerir ligações de dispositivos à rede, recuperar a rede se existirem problemas no seu funcionamento e funcionar como nó de emissão e recepção de dados, [49].

A escolha do canal é feita através de rastreio à rede pelos canais que a rede pode assumir. Se encontrar canais que tenham movimento este canal é removida da lista das possíveis escolhas, este rastreio é feito até encontrar um canal livre para poder comunicar, [50].

Por sua vez a selecção do PAN ID da rede é feita através do envio de um *broadcast* para em cada canal. Cada rede que estiver em cada canal envia a resposta com o PAN ID que está a utilizar. De seguida o coordenador atribui um PAN ID aleatório diferente das redes vizinhas. Este PAN ID pode ser forçado pelo utilizador através da programação do coordenador mas com o cuidado de garantir que duas redes com o mesmo PAN ID não existam no mesmo raio de acção.

O coordenador tem duas formas de permitir que novos dispositivos entrem na rede. Uma forma é estar sempre disponível para receber novos elementos na rede. Outra forma é só permitir entrada de novos elementos num espaço definido de tempo, [50].

Funcionalidades do router (ZR)

O módulo ZR tem como função funcionar como nó de *routing* entre nós da rede e funcionar como nó de emissão e recepção de dados, [49].

Funcionalidades do End-Device (ZED)

O módulo ZED tem como funcionalidade emitir e receber dados. Este módulo tem a particularidade de permitir que o seu funcionamento entre em modo adormecido (*Sleep*). Isto significa que podemos ter este ZDE momentaneamente desligado da rede. A gestão deste modo pode ser feito de várias formas, por controlo de pino dedicado à função, ou por adormecimento cíclico, ou por adormecimento cíclico com gestão de pino dedicado. Com isto, este modulo permite uma gestão energética mais eficiente visto que só funcionará quando necessário.

A1.2. Topologias

Na implementação de uma rede Zig-Bee, podemos ter várias topologias a aplicar para estruturar uma rede. Uma rede Zig-Bee pode ficar formada em Ponto-a-Ponto, em estrela, em árvore e *Mesh* como ilustra a figura A1.2.

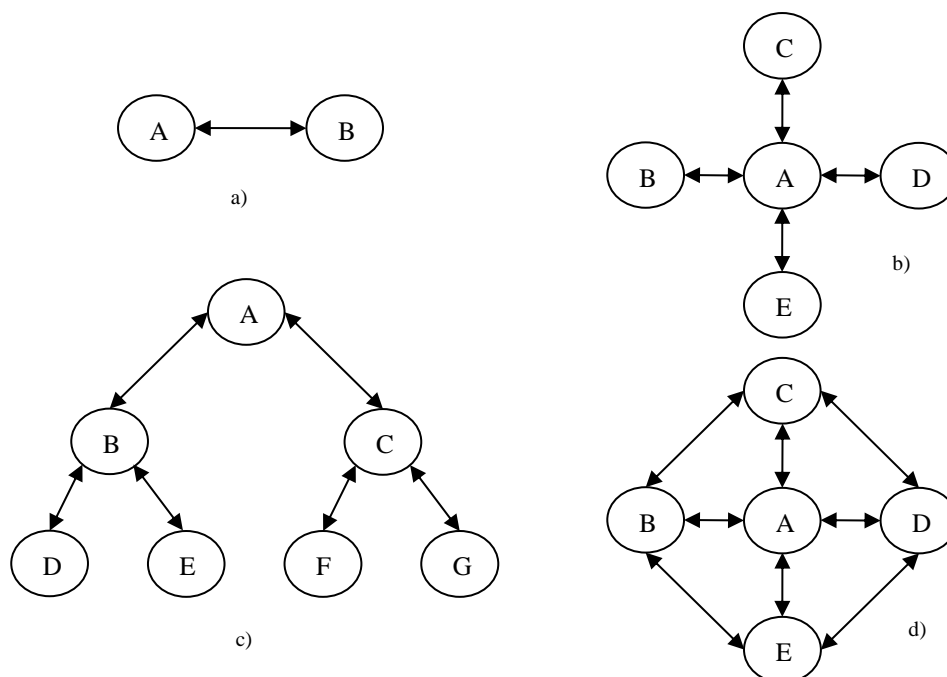


Figura A1.2 - Topologias possíveis em uma rede Zig-Bee.

a) Rede ponto-a-ponto, b) Rede estrela, c) Rede em árvore, d) Rede em *Mesh*

Em uma topologia de ponto-a-ponto temos que a comunicação é feita somente entre dois nós, o permite algoritmos de comunicação mais simples. Em um sistema de iluminação este tipo de topologia será utilizada em situações em que queremos contactar uma luminária directamente.

Em uma topologia em estrela todos os pacotes passam pelo nó central, que faz o endereçamento para o nó de destino.

Em uma topologia em árvore podemos aplicar o conceito de patente, ou seja, os nós superiores acabam por ter uma maior significância na estrutura. Nos sistemas de gestão de iluminação esta topologia poderá ser implementada quando temos modos de gestão locais em ligação com um nó de gestão centralizada.

Em uma topologia *Mesh*, a estrutura é relativamente semelhante à da topologia em árvore com a principal diferença que podemos ter interligação entre nós. Nos sistemas de gestão de iluminação, esta topologia poderá ser utilizada se houver a necessidade de as luminárias instaladas interajam para além das luminárias circundantes.

No âmbito desta tese um misto entre a topologia ponto-a-ponto e a topologia em estrela foi usado, como será descrito no capítulo do dimensionamento.

A1.3. Modos de operação

Existem 5 modos de operação de cada módulo de comunicação. Os pontos seguintes descrevem o que cada modo efectua.

Modo de espera

Quando um módulo não está a efectuar nenhuma operação entra neste modo. Somente sairá deste modo quando receber dados no *buffer* para enviar, ou quando receber dados via meio RF, ou quando entrar em modo de *Sleep* (somente os dispositivos tipo ZED entram neste modo) ou entrar em modo de comando AT, [49].

Modo de transmissão

Quando recebe dados pela porta série o módulo trata os dados e prepara-os para enviar via RF. É garantido que toda a informação necessária para o envio está correcta e em seguida procede ao envio dos dados. Um exemplo de verificação será se os dados contêm a identificação do receptor dos dados, [49].

Modo de recepção

Quando recebe dados através de RF, este reenvia através da porta série para o micro controlador.

Modo de comando AT

Este modo é bastante útil quando temos de configurar o módulo de comunicação remotamente. Este modo permite uma sequência de dados, que permite por exemplo, mudar o destinatário do pacote. O pacote terá de ter a seguinte estrutura, (figura A1.3),

AT	Comando	Valor	CR
----	---------	-------	----

Figura A1.3 - Estrutura do pacote com um comando AT.

Para entrar neste modo, o primeiro comando a ser enviado que terá de ser enviado será “+++”, e para terminar será usado o comando “ATCN” ou esperar o tempo definido no parâmetro *Command Time Out*. A lista de comandos possíveis está disponível em, [4]. Tomando como exemplo a mudança do raio de *broadcast* para 1 a sequência de comandos a enviar será,

+++

ATBH 01

AC

ATCN

Onde o AC será para actualizar e gravar os novos parâmetros.

O módulo Zig-Bee responderá um “OK” se os comandos AT forem válidos ou envia uma mensagem de erro se não o forem e sairá do modo de comandos AT.

Modo de Sleep

Este modo de funcionamento é bastante importante a nível de consumo, mas só poderá ser executado por ZED's. Este modo consiste em que o módulo de comunicação fique suspenso quando não é necessário, minimizando assim o consumo. Este processo foi explicado na definição de ZED no ponto da estrutura da rede.

A1.4. Modos de comunicação

Existem dois tipos de modos de comunicação possíveis com o Zig-Bee, um chamado modo transparente (AT) e outro a *Application Programming Interface* (API).

A principal diferença entre elas é que no modo API, os dados têm de estar formatados de uma certa forma para serem enviados ou recebidos, como por exemplo o uso da ZCL só pode ser no modo API.

No caso do modo AT o que é enviado é definido pelo utilizador, ou seja o módulo de comunicação Zig-Bee faz somente de gateway entre a porta série e o meio físico do Zig-Bee.

No âmbito da tese em questão o modo AT foi o escolhido, visto se ter criado um protocolo para os dados a enviar e a receber, e daí não se ter usado também a ZCL.

AT

Como foi referido anteriormente no modo AT, faz somente a passagem dos dados que se pretende comunicar. Quando o módulo recebe dados no seu *buffer*, este envia-os para o meio físico da mesma forma como os recebeu. Isto permite um interface bastante simples entre o micro controlador e o módulo de comunicação.

API

Outra forma de efectuar uma comunicação será usando o modo API. Este modo define como o pacote a enviar ou a receber terá de estar definido. Existem dois modos de operação dentro do modo API.

No modo API 1 e API 2 a estrutura do pacote será definida consoante a figura A1.4,

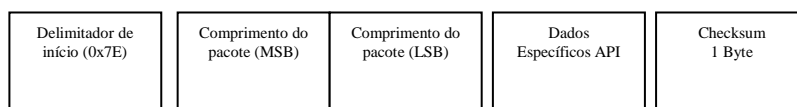


Figura A1.4 - Topologias em uma rede Zig-Bee em modo API.

A diferença entre os modos é que no modo 2 podem existir caracteres que necessitem de ser suprimidos devido à comunicação UART. Por exemplo 0x11 que é usado como XON.

Por exemplo no modo API=1 temos que se pretendemos enviar 0x11 e 0x34 como dados, o pacote a enviar fica da seguinte forma:

0x7E 0x00 0x02 0x11 0x34 0xC8

No modo API=2 se pretendêssemos enviar a mesma informação, o pacote ficaria:

0x7E 0x00 0x02 0x7D 0x31 0x34 0xE7

Quando se pretende omitir um caracter substitui-se esse dado por 0x7D mais um parâmetro que feito o xor desse parâmetro com 0x20 resulta no caracter a omitir, como representado pelo sublinhado no pacote com API=2.

O *checksum* é calculado pela subtração de 0xFF com o somatório dos dados. Existem vários resultados com uso deste tipo de pacote. Uma lista de possibilidades existe em, [50].

A1.5. Endereço de um nó na rede

O endereçamento em uma rede Zig-Bee pode ser feito de duas formas, uma pelo *MAC address* outra por um endereço definido pela rede quando um nó se agrega à rede.

O *MAC Address* é um endereço de 8 *bytes* (64 bits) que é definido nos 3 primeiros *bytes* com o código da Organizational Unique Identifier (OUI), [4] e os restantes 5 *bytes* são definidos pelo fabricante, [47]. Isto garante que nenhum nó terá o mesmo *MAC Address*, [50].

Outro tipo de endereço, que é definido quando o nó entra na rede é um endereço de 2 *bytes* (16 bits). Este endereço tem uma grande vantagem relativamente ao anterior, visto que ocupa menos *bytes* que podem ser usados nos dados. Para determinar o endereço a dar a um nó, são usados um dos seguintes algoritmos. O primeiro, usado no Zig-Bee, é o *Child Skip (CSkip)*. O segundo, usado no Zig-Bee PRO, é o processo estocástico, ou seja random.

O processo *CSkip* consiste em determinar o endereço através de um modelo matemático que tem em conta o uso de três parâmetros, sendo eles o *maxDepth*, *maxChildren* e o *maxRouters*.

O *maxDepth* define o número máximo de profundidade da rede. O *maxChildren* é o número máximo de nós que um router pode ter. O *maxRouters* é o número máximo de routers da rede.

Com este modelo matemático consegue-se dar a cada nó que entre na rede um endereço de 2 *bytes*.

Por sua vez o processo estocástico (random) consiste em o nó que se agrega a uma rede define o seu próprio endereço e através de um envio em *broadcast* anuncia o seu endereço, e se nenhum nó da rede não tiver esse endereço este guarda o endereço que gerou inicialmente. Se um nó já tiver este endereço o nó que enviou o *broadcast* é notificado e gera outro endereço e repete o processo até que adquira um endereço que ainda não esteja agregado à rede.

A1.6. Routing

Existem várias formas de executar o *routing* em uma rede, mas em uma rede Zig-Bee existe o *broadcasting*, o *route* em *mesh* e o *route* em árvore.

Um conceito que tem de ser introduzido devido à sua importância no processo de *routing*, é o conceito de multi-hop. Multi-hop é um processo de passagem de pacotes de dispositivos em dispositivos para que o pacote chegue ao destinatário. Este processo permite que barreiras físicas ou falhas em dispositivos na rede não apresentem uma falha severa na rede.

Route em Broadcast

O route em broadcast permite que todos os nós da rede recebam os dados enviados por um só nó. Ou seja, independentemente do tamanho da rede temos que o pacote enviado por um nó passa por todos os nós. Ou seja, de nó em nó vão replicando o pacote recebido, dentro do raio de *broadcast* definido, até abranger toda a rede. Este tipo de *route* apresenta uma desvantagem, isto porque o emissor não tem a confirmação de recepção o que pode fazer com que hajam falhas na rede que não sejam detectados.

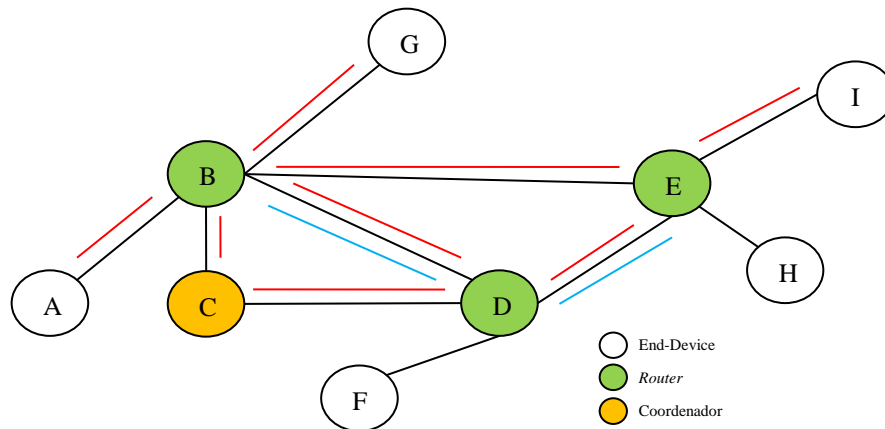
O raio de *broadcast* referenciado anteriormente é um detalhe que precisa de ser analisado cuidadosamente. Como uma comunicação em *broadcast* consome bastante largura de banda e não é a forma mais eficiente de comunicar, [48], o raio correctamente definido pode otimizar o funcionamento da rede. Isto porque se tivermos um raio definido para dois, isto significa que só existirá dois hops entre routers próximos, isto porque só routers e coordenadores replicam o *broadcast*, [1]. Este tipo de *broadcast* é muito semelhante a um *multicast*. O *route* em *broadcast* permite um máximo de 30 hops com um módulo Zig-Bee PRO, [47].

Route em Mesh

Por sua vez um *route* numa rede em *mesh* é bastante eficaz no que diz respeito a tempos e a ocupação do canal. Se tivermos uma rede em que um nó necessitar de comunicar com outro nó que não está ao seu alcance, e apresenta vários obstáculos no seu caminho a informação chegará ao seu destino, sendo isto conseguido é conseguido com o uso de uma tabela de *routing*, [47].

No arranque da rede nenhuma informação sobre a localização dos nós pertencentes à rede e nenhuma informação de *routing*, estão será criada uma tabela de *routing* para cada nó. Esta tabela é criada com um processo denominado por *route discovery process*, [47]. Este processo consiste em o nó que pretende comunicar envia um pacote em *broadcasting* que tem como função descobrir o caminho mais forte até ao nó destinatário. Nesta fase qualquer *router* é um possível caminho na rede. A cada *hop* no caminho é dado um valor que determina a qualidade do *hop*, [47]. Este valor, que pode ir de 1 a 7, determina se o caminho é valido ou não. Se existirem duas possibilidades de *route* e um tiver o valor 1 e outro o valor 5, o caminho escolhido será o que apresentar o valor 1 visto que apresenta a melhor possibilidade de *route*. Isto significa que é o caminho que apresenta menos possibilidades de falhas no seu percurso. O nó destinatário tem um tempo de espera até que receba todas as possibilidades de *routing* disponíveis, [47].

Na figura A1.5 é apresentada uma rede, onde o nó A quer comunicar com o nó I está representado o processo de descobrir o melhor caminho de *routing* entre nós.

Figura A1.5 - Esquema para descobrir o melhor *routing* em *mesh*.

Como podemos observar na figura, as linhas a vermelho representam o *broadcast* enviado pelos nós. Para mostrar como é feita a escolha do caminho será escolhido somente duas possibilidades de entre todas as possíveis. A primeira possibilidade será a passagem entre o nó A->B->E->I e a segunda possibilidade A->B->D->E->I. A tabela seguinte mostra os pesos de cada *hop* entre os nós.

Caminho 1		Caminho 2	
Movimento	Peso	Movimento	Peso
A->B	1	A->B	1
B->E	4	B->D	2
E->I	1	D->E	1
		E->I	1

Tabela A1.1 - Representação dos pesos de *routing*.

O caminho escolhido será o caminho 2, isto porque apresenta menor peso nos *hop's* realizados, o que significa que é a melhor opção para se atingir uma comunicação sem falhas. Tira-se também a conclusão que o caminho escolhido nem sempre é o mais curto mas sim o que tiver melhor classificação, [47].

O nó emissor ficará a saber qual o melhor caminho a fazer, recebendo a resposta do router a qual pertence o nó destinatário. Esta comunicação é realizada em *unicast* conforme representado a azul na figura acima, [47].

Quando existe uma falha na rede, é enviado um pacote com informação de erro e um novo processo de descoberta de melhor *route* é efectuado, [48].

São permitidos até 30 *hops* numa *routing* em *mesh* utilizando um módulo Zig-Bee PRO.

Route em Árvore

O *route* em árvore é bastante semelhante ao *route* em mesh com a principal desvantagem que se houver uma falha de um nó superior temos uma falha desse ponto da rede para baixo, [47].

Anexo B - Software

Classe info_trama

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Controlo_de_luminarias
{
    class info_trama
    {

        string emissora;
        string receptora;
        string codigo;
        string valor;
        string CRC;

        // Construtor
        public info_trama()
        {
            emissora = "";
            receptora = "";
            codigo = "";
            valor = "";
            CRC = "";
        }

        // Destrutor
        ~info_trama()
        {
        }

        //Afecta os valores
        public void _Receptora(string recep) { receptora = recep; }

        public void _Emissora(string emi) { emissora = emi; }

        public void _Codigo(string cod) { codigo = cod; }

        public void _Valor(string val) { valor = val; }

        public void _cRC(string c) { CRC = c; }

        //Devolve valores
        public string _receptora(){return receptora;}

        public string _emissora(){return emissora;}

        public string _codigo(){return codigo;}

        public string _valor(){return valor;}

        public string _CRC(){return CRC;}

        public void descripta_trama(string pacote)
        {
            int ini = 0;
            int fim = 0;
            int comp = 0;

            ini = pacote.IndexOf('&',ini);
            fim= pacote.IndexOf('&', ini +1);
            comp = fim - ini - 1;
            emissora = pacote.Substring(ini+1, comp);

            ini = pacote.IndexOf('&', fim);
            fim = pacote.IndexOf('&', ini + 1);
            comp = fim - ini - 1;
            receptora = pacote.Substring(ini + 1, comp);

            ini = pacote.IndexOf('&', fim);
            fim = pacote.IndexOf('&', ini + 1);
```

```

        comp = fim - ini - 1;
        codigo = pacote.Substring(ini + 1, comp);

        ini = pacote.IndexOf('&', fim);
        fim = pacote.IndexOf('&', ini + 1);
        comp = fim - ini - 1;
        valor = pacote.Substring(ini + 1, comp);

        ini = pacote.IndexOf('&', fim);
        fim = pacote.IndexOf('&', ini + 1);
        comp = fim - ini - 1;
        CRC = pacote.Substring(ini + 1, comp);
    }

    //Validações
    public bool valida_CRC()
    {
        try
        {
            if ((Convert.ToInt32(_receptora()) + (Convert.ToInt32(_emissora()) + Convert.ToInt32(_codigo()) + Convert.ToInt32(_valor()))
            == Convert.ToInt32(_CRC()))
            {
                return true;
            }
        }
        catch (Exception _e) { return false; }

        return false;
    }

    //Geradores
    public string gera_trama_enviar(string receptora, string emissora, string codigo, string valor, string CRC)
    {
        return ("&" + receptora + "&" + emissora + "&" + codigo + "&" + valor + "&" + CRC + "&");
    }

    public string gera_CRC(string receptora, string emissora, string codigo, string valor)
    {
        return Convert.ToString(Convert.ToInt32(receptora) + Convert.ToInt32(emissora) + Convert.ToInt32(codigo) +
        Convert.ToInt32(valor));
    }

    public string des_receptora(string pacote)
    {
        string dados = "";

        int ini = pacote.IndexOf('&');
        int fim = pacote.IndexOf('&', ini + 1);

        //numero da luminaria
        for (int i = 0; i < 3; i++)
            dados += pacote[i + 1];

        emissora = dados;
        return dados;
    }

    public string des_emissora(string pacote)
    {
        string dados = "";

        //numero da luminaria
        for (int i = 0; i < 3; i++)
            dados += pacote[i + 5];

        receptora = dados;
        return dados;
    }

    public string des_codigo(string pacote)
    {
        string dados = "";

        //numero da luminaria

```

```

        for (int i = 0; i < 2; i++)
            dados += pacote[i + 9];

        codigo = dados;
        return dados;
    }

    public string des_valor(string pacote)
    {
        string dados = "";

        for (int i = 0; i < 4; i++)
            dados += pacote[i + 12];

        valor = dados;
        return dados;
    }

    public string des_CRC(string pacote)
    {
        string dados = "";

        //numero da luminaria
        for (int i = 0; i < 4; i++)
            dados += pacote[i + 17];

        CRC = dados;
        return dados;
    }

    public void preenche_dados(string pacote)
    {
        emissora = des_codigo(pacote);
        receptora = des_receptora(pacote);
        codigo = des_codigo(pacote);
        valor = des_valor(pacote);
        CRC = des_CRC(pacote);
    }
}

```

Form1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.IO.Ports;
using System.IO;
using System.Threading;

namespace Controlo_de_luminarias
{
    public partial class Form1 : Form
    {
        SerialPort porta = new SerialPort();
        string trama_enviar = "";
        const string n_central_luminaria = "000";

        //Comunicação (Receive)
        info_trama trama;
        string trama_auxiliar = "";
        int count = 0;
        ///////////////////////////////////

        //Variaveis globais
        int circundantes = 0;
    }
}

```

```
public Form1()
{
    InitializeComponent();
    porta.DataReceived += new SerialDataReceivedEventHandler(port_DataReceived);
    _default(8,6,5,3,1,3);
}

// Funções Auxiliares //////////////////////////////////////

// Prepara os parametros da porta COM
private void _default(int com, int baud, int bit_dados, int pari, int bit_paragem, int c_fluxo)
{
    com_list.SelectedItem = com_list.Items[com];
    baud_rate.SelectedItem = baud_rate.Items[baud];
    bits_dados.SelectedItem = bits_dados.Items[bit_dados];
    paridade.SelectedItem = paridade.Items[pari];
    bits_paragem.SelectedItem = bits_paragem.Items[bit_paragem];
    cont_fluxo.SelectedItem = cont_fluxo.Items[c_fluxo];
}

//Efectua acções conforme o código da trama recebida
private void executa_codigo(info_trama_trama)
{
    if (_trama_codigo() == "1")
    {
        if (trama_valor() == "1111")
        {
            Escrever("\n");
            Escrever("Resposta da luminária: Pedido efectuado");
            Escrever("\n");
        }
        else if (trama_valor() == "0000")
        {
            Escrever("\n");
            Escrever("Resposta da luminária: Pedido não efectuado");
            Escrever("\n");
        }
    }
    if (_trama_codigo() == "2")
    {
        Escrever_lum(_trama_valor());
    }
    if (_trama_codigo() == "3")
    {
        Escrever_temp(_trama_valor());
    }
    if (_trama_codigo() == "4")
    {
        Escrever_tractus(_trama_valor());
    }
    if (_trama_codigo() == "5")
    {
        Escrever_pwm(_trama_valor());
    }
    if (_trama_codigo() == "9")
    {
        Escrever_MAX_SUN(_trama_valor());
    }
    if (_trama_codigo() == "10")
    {
        Escrever_MAX_TEMP(_trama_valor());
    }
    if (_trama_codigo() == "12")
    {
        Escrever_leitura_gestao(_trama_valor() + "\n");
    }
    if (_trama_codigo() == "15")
    {
        switch (_trama_valor())
        {
            case "0":
                Escrever_MODAL_FUNCIONAMENTO("WAIT");
                break;
            case "1":
                Escrever_MODAL_FUNCIONAMENTO("TRACTUS");
                break;
        }
    }
}
```



```

        case "2":
            Escrever_MODO_FUNCIONAMENTO("SUN");
            break;
        case "3":
            Escrever_MODO_FUNCIONAMENTO("TRACTUS_AUX");
            break;
        case "4":
            Escrever_MODO_FUNCIONAMENTO("PROTECT");
            break;
    }
}
}

////////////////////////////////////

private void botao_ligar_Click(object sender, EventArgs e)
{
    if(botao_ligar.Text == "Ligar")
    {
        try
        {
            porta.PortName = com_list.Text;
            porta.BaudRate = int.Parse(baud_rate.Text);
            porta.DataBits = int.Parse(bits_dados.Text);

            //definição da paridade
            if (paridade.Text == "Par")
            {
                porta.Parity = Parity.Even;
            }
            else if (paridade.Text == "Impar")
            {
                porta.Parity = Parity.Odd;
            }
            else if (paridade.Text == "Nenhum")
            {
                porta.Parity = Parity.None;
            }
            else if (paridade.Text == "Marca")
            {
                porta.Parity = Parity.Mark;
            }
            else if (paridade.Text == "Espaço")
            {
                porta.Parity = Parity.Space;
            }
        }
        catch (Exception e1_)
        {
            texto.Text = "Verifique os valores!";
            Escrever_cor("Verifique os Valore!", Color.Red);
        }
        try
        {
            porta.Open();
            tab.TabPages[1].BringToFront();
            texto.Text = "Porta " + com_list.Text + " aberta com sucesso, com as seguintes características:";
            Escrever_cor("Porta " + com_list.Text + " aberta com sucesso, com as seguintes características:", Color.Green);
            trama = new info_trama();

            botao_ligar.Text = "Desligar";
        }
        catch (Exception _e)
        {
            texto.Text = "Porta " + com_list.Text + " não foi aberta com sucesso!";
        }
    }

    else if (botao_ligar.Text == "Desligar")
    {
        try
        {
            porta.Close();
        }
    }
}

```

```
        texto.Text = "Porta " + com_list.Text + " fechada com sucesso!";
        botao_ligar.Text = "Ligar";
    }
    catch (Exception _e)
    {
        texto.Text = "Porta " + com_list.Text + " não foi fechada com sucesso!";
    }
}

private void sairToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Tem a certeza que pretende fechar a aplicação?",
"Fechar", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    porta.Close();
    Close();
}

private void sobreToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Criado por Ricardo Lares");
}

private void control_luz_Scroll(object sender, EventArgs e)
{
    valor_intensidade.Text = Convert.ToString(control_luz.Value) + "%";
    temporizador.Interval = 500;
    temporizador.Enabled = true;
}

//Handler para ler os dados recebidos no PORT
private void port_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    if (!porta.IsOpen) return;

    int bytes = porta.BytesToRead;
    byte[] buffer = new byte[bytes];

    porta.Read(buffer, 0, bytes);

    trama_auxiliar += System.Text.ASCIIEncoding.Default.GetString(buffer);

    for (int k = 0; k < trama_auxiliar.Length; k++)
    {
        if (trama_auxiliar[k] == '&')
            count++;
    }

    if (count != 6)
        count = 0;

    if (count > 6)
    {
        count = 0;
        buffer = null;
    }

    Escrever("\n");
    Escrever("Trama recebida até ao momento: ");
    Escrever(trama_auxiliar);
    Escrever("\n");

    // trama de códigos
    if (trama_auxiliar[trama_auxiliar.Length - 1] == '&' && count == 6)
    {
        Escrever("\n");
        Escrever("Trama recebida: ");
        Escrever(trama_auxiliar);
        Escrever("\n");
        trama.desencrpta_trama(trama_auxiliar);
    }
}
```

```

        if (trama.valida_CRC())
        {
            Escrever("Numero da receptora: ");
            Escrever(trama._emissora());
            Escrever("\n");
            Escrever("Numero da emissora: ");
            Escrever(trama._receptora());
            Escrever("\n");
            Escrever("Numero do código: ");
            Escrever(trama._codigo());
            Escrever("\n");
            Escrever("Valor: ");
            Escrever(trama._valor());
            Escrever("\n");
            Escrever("CRC: ");
            Escrever(trama._CRC());
            Escrever("\n");
            executa_codigo(trama); //Executa os comandos do código recebido
        }
        else
        {
            Escrever("Trama Recebida não está valida");
            Escrever("\n");
        }
    }

    /*
    if (trama.valida_CRC(trama_auxiliar))
    {
        Escrever("CRC correcto!");
        Escrever("\n");
    }
    else
    {
        Escrever("CRC incorrecto!");
        Escrever("\n");
    }

    if(Convert.ToInt16(trama.des_codigo(trama_auxiliar)) == 2)
    {
        Escrever("Código 2!!");
        Escrever("\n");
    }

    */
    trama_auxiliar = "";
    count = 0;

}

/*
// trama de ack
if (trama[trama.Length - 1] == '%' && trama.Length == 8)
{
    if (Convert.ToInt16(des_codigo(trama)) == 2)
    {
        Escrever("Código 2!!");
        Escrever("\n");
    }
}
*/

}

//Escrever nas caixas de texto
private void Escrever(string msg)
{
    texto.Invoke(new EventHandler(delegate
    {
        {
            texto.AppendText(msg);
            texto.ScrollToCaret();
        }
    }));
}

```

```
private void Escrever_leitura_gestao(string msg)
{
    leitura_circundantes_texto.Invoke(new EventHandler(delegate
    {
        //texto.SelectionFont = new Font(rtfTerminal.SelectionFont, FontStyle.Bold);
        //rtfTerminal.SelectionColor = LogsgTypeColor[(int)msgtype];

        leitura_circundantes_texto.AppendText(msg);
        leitura_circundantes_texto.ScrollToCaret();

    }));
}

private void Escrever_cor(string texto, Color cor)
{
    texto1.SelectionFont = new Font(texto1.SelectionFont, FontStyle.Bold);
    texto1.AppendText "[" + System.DateTime.Today.Day.ToString() + "-");
    texto1.AppendText(System.DateTime.Today.Month.ToString() + "-");
    texto1.AppendText(System.DateTime.Today.Year.ToString() + "] ");
    texto1.AppendText(System.DateTime.UtcNow.TimeOfDay.Hours.ToString() + ":");
    texto1.AppendText(System.DateTime.UtcNow.TimeOfDay.Minutes.ToString() + ":");
    texto1.AppendText(System.DateTime.UtcNow.TimeOfDay.Seconds.ToString() + " - ");
    texto1.SelectionFont = new Font(texto1.SelectionFont, FontStyle.Bold);
    texto1.ScrollToCaret();
    texto1.ForeColor = cor;
    texto1.AppendText(texto);
    texto1.ScrollToCaret();
}

private void Escrever_lum(string msg)
{
    texto.Invoke(new EventHandler(delegate
    {
        lum.Text = msg;

    }));
}

private void Escrever_temp(string msg)
{
    texto.Invoke(new EventHandler(delegate
    {
        //texto.SelectionFont = new Font(rtfTerminal.SelectionFont, FontStyle.Bold);
        //rtfTerminal.SelectionColor = LogsgTypeColor[(int)msgtype];

        temp.Text = msg;

    }));
}

private void Escrever_tractus(string msg)
{
    texto.Invoke(new EventHandler(delegate
    {
        //texto.SelectionFont = new Font(rtfTerminal.SelectionFont, FontStyle.Bold);
        //rtfTerminal.SelectionColor = LogsgTypeColor[(int)msgtype];

        sensor_info.Text = msg;

    }));
}

private void Escrever_pwm(string msg)
{
    texto.Invoke(new EventHandler(delegate
    {
        PWM_valor.Text = msg;

    }));
}

private void Escrever_MAX_SUN(string msg)
{
    texto.Invoke(new EventHandler(delegate
```

```

        {
            escrita_sun_max.Text = msg;
        }
    ));
}

private void Escrever_MAX_TEMP(string msg)
{
    texto.Invoke(new EventHandler(delegate
    {
        escrita_temp_max.Text = msg;
    }));
}

private void Escrever_MODALIDADE_FUNCIONAMENTO(string msg)
{
    texto.Invoke(new EventHandler(delegate
    {
        modo_funcionamento_texto.Text = msg;
    }));
}

/// //////////////////////////////////////

private void limparCaixasDeTextoToolStripMenuItem_Click(object sender, EventArgs e)
{
    texto.Clear();
    texto1.Clear();
}

private void fazer_leitura_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("3");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido do sensor de Temperatura");
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}

private void env_comando_Click(object sender, EventArgs e)
{
    string input = env_comando_text.Text;
    string mens = "";
    char[] values = input.ToCharArray();
    string[] men = new string[values.Length];
    byte[] rawBytes = new byte[values.Length];
    int i = 0;
    foreach (char letter in values)
    {
        // Achar o valor decimal do caracter
    }
}

```

```
int value = Convert.ToInt32(letter);
// Converte o valor decimal para o valor hexadecimal
string hexOutput = String.Format("{0:x}", value);
men[i] = hexOutput;

//texto.AppendText(letter + " -> Hexadecimal value:" + men[i] + " Int:" + value + "\n");
i++;
}

for (int j = 0; j < men.Length ; j++)
{
    mens += men[j];
}

try
{
    porta.Write(env_comando_text.Text);
    Escrever("Enviou o comando manual!\n");
}
catch (Exception _e) {
    Escrever("\n");
    Escrever("Não Enviou o comando manual!\n");
}
}

private void fazer_leitura_luminosidade_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("2");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido do sensor de luminosidade");
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}

private void fazer_leitura_movimento_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("4");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido do sensor de luminosidade");
        }
        catch (Exception _e)
        {

```

```
        Escrever("\n");
        Escrever("Não conseguiu enviar o pedido");
    }
}
else
{
    Escrever("\n");
    Escrever("Verifique o numero da luminaria a contactar!");
}
}

private void fazer_leitura_pwm_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("5");
        trama._Valor("0000");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido de leitura de PWM");
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}

private void temporizador_Tick(object sender, EventArgs e)
{
    temporizador.Enabled = false;
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("6");
        double pwmValue = -2.55 * control_luz.Value + 255;
        string pwmValue_string = Convert.ToString(Convert.ToInt16(pwmValue));
        label15.Text = pwmValue_string;
        trama._Valor(pwmValue_string);
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de alteração do PWM");
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}
```

```
// Definição do valor maximo de luminosidade
private void max_lum_bot_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        try
        {
            int lum_max_user = Convert.ToInt16(lum_maxima_text.Text);
            if (lum_max_user < 1001 && lum_max_user > 99)
            {
                //enviar código
                trama._Receptora(n_lum.Text);
                trama._Emissora(n_central_luminaria);
                trama._Codigo("7");
                trama._Valor((Convert.ToInt32(lum_maxima_text.Text)/4).ToString());
                trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
                porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(),
trama._CRC()));
                Escrever("\n");
                Escrever("Envio da definição do valor de luminosidade");
            }
            else
            {
                Escrever("\n");
                Escrever("O valor inserido não é permitido");
            }
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Verifique o tipo de dados que inseriu. Tem de ser um valor inteiro");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}

//Definição do valor máximo de temperatura
private void max_temp_bot_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        try
        {
            int temp_max_user = Convert.ToInt16(temp_maxima_text.Text);
            if (temp_max_user < 100)
            {
                trama._Receptora(n_lum.Text);
                trama._Emissora(n_central_luminaria);
                trama._Codigo("8");
                trama._Valor(temp_maxima_text.Text);
                trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
                porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(),
trama._CRC()));
                Escrever("\n");
                Escrever("Envio da definição do valor da temperatura máxima");
            }
            else
            {
                Escrever("\n");
                Escrever("O valor inserido não é permitido");
            }
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Verifique o tipo de dados que inseriu. Tem de ser um valor inteiro");
        }
    }
    else
    {

```



```
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }

}

//Obtenção das luminárias circundantes
private void obter_luminarias_circundantes_botao_Click(object sender, EventArgs e)
{
    if (n_luminaria_gestao.Text != "")
    {
        trama._Receptora(n_luminaria_gestao.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("12");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido das luminárias circundantes");
            leitura_circundantes_texto.Clear();
            Escrever_leitura_gestao("Luminárias circundantes à luminária numero " + n_luminaria_gestao.Text + ":\n");
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}

//Gravação das luminárias circundantes
private void gravar_luminarias_circundantes_botao_Click(object sender, EventArgs e)
{
    string valor_aux = "";
    if (n_luminaria_gestao.Text != "")
    {
        trama._Receptora(n_luminaria_gestao.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("11");
        for (int k = 0; k < escrita_circundantes_texto.Lines.Length; k++)
        {
            if (escrita_circundantes_texto.Lines[k] != "")
            {
                trama._Valor(escrita_circundantes_texto.Lines[k]);
                trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
                //trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());
                try
                {
                    porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(),
trama._CRC()));
                    Thread.Sleep(1000);
                }
                catch (Exception _e)
                {
                }
            }
        }
        trama._Valor("255");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        //trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());
        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido de gravação das luminárias circundantes");
        }
    }
}
```

```
    }
    catch (Exception _e)
    {
        Escrever("\n");
        Escrever("Não conseguiu enviar o pedido");
    }
}
else
{
    Escrever("\n");
    Escrever("Verifique o numero da luminaria a contactar!");
}
}

//Leitura do valor máximo de SUN
private void leitura_sun_max_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("9");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido de leitura do valor máximo de SUN");
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}

//Leitura do valor máximo de TEMP
private void leitura_temp_max_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("10");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido de leitura do valor máximo de TEMP");
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}
}
```

```
//Limpa registos das luminárias circundantes
private void limpa_registos_circundantes_Click(object sender, EventArgs e)
{
    if (n_luminaria_gestao.Text != "")
    {
        trama._Receptora(n_luminaria_gestao.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("11");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de limpeza dos registos das luminárias circundantes");
            escrita_circundantes_texto.Clear();
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}

private void modo_funcionamento_botao_Click(object sender, EventArgs e)
{
    if (n_lum.Text != "")
    {
        trama._Receptora(n_lum.Text);
        trama._Emissora(n_central_luminaria);
        trama._Codigo("15");
        trama._Valor("0");
        trama._cRC(trama.gera_CRC(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor()));
        trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC());

        try
        {
            porta.Write(trama.gera_trama_enviar(trama._receptora(), trama._emissora(), trama._codigo(), trama._valor(), trama._CRC()));
            Escrever("\n");
            Escrever("Envio o comando de pedido do modo de funcionamento");
            escrita_circundantes_texto.Clear();
        }
        catch (Exception _e)
        {
            Escrever("\n");
            Escrever("Não conseguiu enviar o pedido");
        }
    }
    else
    {
        Escrever("\n");
        Escrever("Verifique o numero da luminaria a contactar!");
    }
}
}
}
```

Código Arduino

```
#include <stdlib.h>
#include <string.h>
#include <EEPROM.h>
```

```
#define N_LUMINARIA 2
#define MAX_CHAR_VAL 20
#define MAX_CHAR_TRAMA 110
#define SUN_MAX_VAL_EEPROM_POS 2
#define TEMP_MAX_VAL_EEPROM_POS 1
#define MAX_LUMINARIAS_CIRCUNDANTES 20
#define DELAY_TIME_COM 50
#define COM_BAUD 38400

//modos de funcionamento
#define WAIT 0 //Modo de repouso da luminária
#define TRACTUS 1 // Modo acionado por tractus
#define SUN 2 //Modo acionado pelo SUN sensor
#define TRACTUS_AUX 3 //Modo acionado por uma luminária circundante
#define PROTECT 4 //Modo de protecção

String n_luminaria = "2";
int MODE = -1;
int ANT_MODE = -1;
boolean TRACTUS_ANT = false;

//Declarações dos pinos a usar
int tempsensorPin = A4;
int luzsensorPin = A2;
int ledPin = 13;
int tractusPin = 2;

//Variáveis valores dos sensores
double tempsensorValue = 0;
double luzsensorValue = 0;
int tempsensorValueDec = 0;
int luzsensorValue1Dec = 0;
int intensidade_luz = 9999;

int max_val_sun = 0;
int max_val_temp = 0;

// Variáveis luz
float lightVoltage = 0;

//Variáveis temperatura
float tempVoltage = 0;
int temperaturaC = 0;
int temperaturaK = 0;
float tempCelsius = 0;

//Variáveis de controlo de PWM
int pwmValue = 255;
int pwmPin = 11;

//Variáveis de tractus
int tractus_val = 0;

//Variáveis de comunicação
byte byteRecebido[MAX_CHAR_TRAMA];

char str[MAX_CHAR_TRAMA];
int countBytes = 0;
String trama_enviar;

char emissora[MAX_CHAR_VAL];
char receptora[MAX_CHAR_VAL];
char codigo[MAX_CHAR_VAL];
char valor[MAX_CHAR_VAL];
char CRC[MAX_CHAR_VAL];

//Variáveis de gestão
int count_gestao = 10;
int luminarias_gestao[MAX_LUMINARIAS_CIRCUNDANTES][2];

// Funções auxiliares
```

```

void descripta_trama(char trama[]){
    int count = 0;
    int pos = 0, pmissora = 0, preceptora = 0, pcodigo = 0, pvalor = 0, pCRC = 0;

    while(pos < MAX_CHAR_TRAMA){
        if(trama[pos] == '&'){count++;}

        if(trama[pos] != '&')
        {
            if(count == 1)
            {
                sprintf(&receptora[preceptora], "%c", trama[pos]);
                preceptora++;
            }
            else if(count == 2)
            {
                sprintf(&emissora[pmissora], "%c", trama[pos]);
                pmissora++;
            }
            else if(count == 3)
            {
                sprintf(&codigo[pcodigo], "%c", trama[pos]);
                pcodigo++;
            }
            else if(count == 4)
            {
                sprintf(&valor[pvalor], "%c", trama[pos]);
                pvalor++;
            }
            else if(count == 5)
            {
                sprintf(&CRC[pCRC], "%c", trama[pos]);
                pCRC++;
            }
        }
        pos++;
    }
}

int cria_CRC(char emi[MAX_CHAR_VAL], char recep[MAX_CHAR_VAL], char cod[MAX_CHAR_VAL], char
val[MAX_CHAR_VAL]){
    for(int k=0;k<MAX_CHAR_VAL;k++){
        CRC[k] = 0;
    }
    return (atoi(emi) + atoi(recep) + atoi(cod) + atoi(val));
}

boolean valida_CRC(char emi[MAX_CHAR_VAL], char recep[MAX_CHAR_VAL], char cod[MAX_CHAR_VAL], char
val[MAX_CHAR_VAL], char_crc[MAX_CHAR_VAL]){
    //int result = (int)cria_CRC(emi,recep,cod,val)

    if((atoi(emi) + atoi(recep) + atoi(cod) + atoi(val)) == atoi(_crc))
        return true;
    else
        return false;
}

String cria_trama(char recep[MAX_CHAR_VAL], char emi[MAX_CHAR_VAL], char cod[MAX_CHAR_VAL], char
val[MAX_CHAR_VAL], char_crc[MAX_CHAR_VAL]){
    return "&" + String(recep) + "&" + String(emi) + "&" + String(cod) + "&" + String(val) + "&" + String(_crc) + "&";
}

void blinkled(int j){
    for(int i=0; i<j;i++){
        digitalWrite(ledPin, LOW);
        delay(200);
        digitalWrite(ledPin, HIGH);
        delay(200);
    }
}

void copia_array(char array_1[MAX_CHAR_VAL], char array_2[MAX_CHAR_VAL]){
    for(int i = 0; i < MAX_CHAR_VAL; i++){
        array_1[i] = array_2[i];
    }
}

```

```
}  
}  
  
void troca_array(char array_1[MAX_CHAR_VAL], char array_2[MAX_CHAR_VAL]){  
    char _aux[MAX_CHAR_VAL];  
  
    copia_array(_aux,emissora);  
    copia_array(emissora,receptora);  
    copia_array(receptora,_aux);  
}  
  
void limpa_array(char arr[MAX_CHAR_VAL])  
{  
    for(int y = 0; y < MAX_CHAR_VAL; y++){  
        arr[y] = ' ';  
    }  
}  
  
// Funções dos códigos das tramas ///////////////////////////////////  
  
//Comandos AT  
void codigo0(){blinkled(1);}   
  
//ACK  
void codigo1(){  
    blinkled(1);  
}  
  
//Leitura do sensor de Luz  
void codigo2(){  
    troca_array(emissora, receptora);  
  
    luzsensorValue = analogRead(luzsensorPin);  
    lightVoltage = luzsensorValue * 5 / 1024;  
    delay(1);  
  
    sprintf(valor,"%d", lightVoltage);  
    String(cria_CRC(receptora,emissora,codigo,valor)).toCharArray(CRC, MAX_CHAR_VAL);  
    blinkled(2);  
}  
  
//Leitura do sensor de Temperatura  
void codigo3(){  
    troca_array(emissora, receptora);  
  
    tempsensorValue = analogRead(tempsensorPin);  
    delay(1);  
    String((int)tempsensorValue).toCharArray(valor,MAX_CHAR_VAL);  
    String(cria_CRC(receptora,emissora,codigo,valor)).toCharArray(CRC, MAX_CHAR_VAL);  
    blinkled(3);  
}  
  
//Leitura do tractus sensor pedido por comunicação  
void codigo4(){  
    troca_array(emissora, receptora);  
    int tractus = digitalRead(2);  
    delay(1);  
    String(tractus).toCharArray(valor,MAX_CHAR_VAL);  
    String(cria_CRC(receptora,emissora,codigo,valor)).toCharArray(CRC, MAX_CHAR_VAL);  
    blinkled(4);  
}  
  
//Leitura do valor de PWM (Feito)  
void codigo5(){  
    troca_array(emissora, receptora);  
    sprintf(valor,"%d", pwmValue);  
    sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));  
    blinkled(5);  
}  
  
//Escrita do valor de PWM atraves de comunicação (Feito)  
void codigo6(){  
    troca_array(emissora, receptora);  
    pwmValue = atoi(valor);  
    //sprintf(valor,"%d",pwmValue);  
    analogWrite(pwmPin,pwmValue);  
}
```

```

delay(1);

//envio do ACK
sprintf(codigo,"%d",1);
if(atoi(valor) == pwmValue){sprintf(valor,"%d",1111);}
else {sprintf(valor,"%d",0000);}
sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));

//blinkled(6);
}

void codigo7()//Escrita do valor máximo de SUN (FEITO)
{
    troca_array(emissora, receptora);
    EEPROM.write(SUN_MAX_VAL_EEPROM_POS,atoi(valor));

    if(EEPROM.read(SUN_MAX_VAL_EEPROM_POS) == atoi(valor)){sprintf(valor,"%d",1111);}
    else {sprintf(valor,"%d",0000);}
    sprintf(codigo,"%d",1);
    sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
    blinkled(7);
}

void codigo8()//Escrita do valor máximo de TEMP(FEITO)
{
    troca_array(emissora, receptora);
    EEPROM.write(TEMP_MAX_VAL_EEPROM_POS,atoi(valor));

    if(EEPROM.read(TEMP_MAX_VAL_EEPROM_POS) == atoi(valor)){sprintf(valor,"%d",1111);}
    else {sprintf(valor,"%d",0000);}
    sprintf(codigo,"%d",1);
    sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
    blinkled(8);
}

void codigo9()//Leitura do MAX de SUN (FEITO)
{
    troca_array(emissora, receptora);
    sprintf(valor,"%d",EEPROM.read(SUN_MAX_VAL_EEPROM_POS)*4);
    sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
    blinkled(9);
}

void codigo10()//Leitura do MAX de TEMP (FEITO)
{
    troca_array(emissora, receptora);
    sprintf(valor,"%d",EEPROM.read(TEMP_MAX_VAL_EEPROM_POS));
    sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
    blinkled(10);
}

void codigo11()//Escrita das luminarias circundantes (FEITO)
{
    troca_array(emissora, receptora);

    if(atoi(valor) == 0){
        for(int h = 10; h < MAX_LUMINARIAS_CIRCUNDANTES + 10; h++){
            EEPROM.write(h,255);
            count_gestao = 10;
            luminarias_gestao[h-10][0] = 0;
        }
        sprintf(codigo,"%d",1);
        sprintf(valor,"%d",1111);
        sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
        Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
        delay(DELAY_TIME_COM);
    }
    else if(atoi(valor) != 255){
        EEPROM.write(count_gestao,atoi(valor));
        luminarias_gestao[count_gestao - 10][0] = atoi(valor);
        //luminarias_gestao[count_gestao-10] = atoi(valor);
        count_gestao++;
    }
    else{
        sprintf(codigo,"%d",1);
        sprintf(valor,"%d",1111);
    }
}

```

```

    sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
    Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
    delay(DELAY_TIME_COM);
}

}

void codigo12()//Leitura das luminarias circundantes (FEITO)
{
    troca_array(emissora, receptora);
    int valor_aux = 0;

    for(int h = 10; h < MAX_LUMINARIAS_CIRCUNDANTES + 10; h++){
        valor_aux = EEPROM.read(h);
        if(valor_aux != 255){
            sprintf(valor,"%d",valor_aux);
            sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
            Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
            delay(5*DELAY_TIME_COM);
        }
    }
}

//Envia para as luminarias circundantes a informação para entrarem em modo TRACTUS_AUX
void envia_tractus_sinal(void){
    for(int k = 0; k < MAX_LUMINARIAS_CIRCUNDANTES; k++){
        if(luminarias_gestao[k][0] != 0){
            sprintf(emissora,"%d",1);
            sprintf(receptora,"%d",luminarias_gestao[k][0]);
            sprintf(codigo,"%d",16);
            sprintf(valor,"%d",1);
            sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
            Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
            delay(5*DELAY_TIME_COM);
        }
    }
}

void envia_tractus_off_sinal(void){
    for(int k = 0; k < MAX_LUMINARIAS_CIRCUNDANTES; k++){
        if(luminarias_gestao[k][0] != 0){
            sprintf(emissora,"%d",1);
            sprintf(receptora,"%d",luminarias_gestao[k][0]);
            sprintf(codigo,"%d",17);
            sprintf(valor,"%d",0);
            sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
            Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
            delay(5*DELAY_TIME_COM);
        }
    }
}

//Faz a leitura do SUN sensor e devolve o numero de lux
int leitura_sun(){

    luzsensorValue = analogRead(luzsensorPin);
    lightVoltage = luzsensorValue * 5 / 1024;
    delay(1);

    return luzsensorValue;
}

//Faz a leitura do TEMP sensor e devolve o numero de graus Celsius
int leitura_temp(){

    /*
    tempsensorValue = analogRead(tempsensorPin);
    delay(1);*/

    return tempsensorValue;
}

// Configuração //////////////////////////////////
void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(COM_BAUD);

```



```

analogReference(DEFAULT);
digitalWrite(ledPin, HIGH);
//attachInterrupt(1, tractus_int, RISING);
analogWrite(pwmPin,210);
delay(1);
max_val_sun = EEPROM.read(SUN_MAX_VAL_EEPROM_POS);
delay(1);
max_val_temp = EEPROM.read(TEMP_MAX_VAL_EEPROM_POS);
delay(1);
MODE = WAIT;

for(int h = 0; h < MAX_LUMINARIAS_CIRCUNDANTES ; h++){
    if(EEPROM.read(h +10) != 255){
        luminarias_gestao[h][0] = EEPROM.read(h+10);
        luminarias_gestao[h][1] = 0;
    }
}

delay(3000);
}

void tractus_int()
{
    //pwmValue = 0;
    delay(20);
    blinked(3);
    tractus_val = digitalRead(tractusPin);

    if(tractus_val == 1 && MODE == SUN)
    {
        MODE = TRACTUS;
    }
    else if(tractus_val == 0 && MODE == TRACTUS){
        MODE = SUN;
    }
}

void loop() {

    /////////// COMUNICACAO//////////

    if(leitura_temp() > max_val_temp){
        MODE = PROTECT;
    }

    switch(MODE){
        case PROTECT:
            if(leitura_temp() <= max_val_temp){
                MODE = WAIT;
            }
            break;

        case SUN:
            if(leitura_sun() > 0){
                MODE = WAIT;
            }
            else{
                tractus_val = digitalRead(tractusPin);
                delay(1);
                if(tractus_val == 1 && MODE == SUN)
                {
                    MODE = TRACTUS;
                }
                else if(tractus_val == 0 && MODE == TRACTUS){
                    MODE = SUN;
                }
            }
            break;

        case TRACTUS:
            if(!digitalRead(tractusPin) && leitura_sun() == 0){
                MODE = SUN;
            }
            break;

        case WAIT:

```

```

if(leitura_sun() == 0){
    MODE = SUN;
}
break;

case TRACTUS_AUX:
    //if(leitura_sun() <= max_val_sun){
    // MODE = WAIT;
    //}

    break;

}

if( MODE != ANT_MODE){

    if(MODE == WAIT){
        //ANT_MODE = MODE;
        pwmValue = 255;
        //analogWrite(pwmPin,pwmValue);
    }
    else if(MODE == SUN){
        //ANT_MODE = MODE;
        if(ANT_MODE == TRACTUS){
            envia_tractus_off_sinal();
        }
        pwmValue = 128;
        //analogWrite(pwmPin,pwmValue);
    }
    else if(MODE == TRACTUS){
        //ANT_MODE = MODE;
        pwmValue = 0;
        analogWrite(pwmPin,pwmValue);
        envia_tractus_sinal();
    }
    else if(MODE == PROTECT){
        //ANT_MODE = MODE;
        pwmValue = 255;
        //analogWrite(pwmPin,pwmValue);
    }
    else if(MODE == TRACTUS_AUX){
        //ANT_MODE = MODE;
        pwmValue = 64;
        //analogWrite(pwmPin,pwmValue);
    }
    analogWrite(pwmPin,pwmValue);
    ANT_MODE = MODE;

}

//COMUNICACAO
while(Serial.available() > 0){
    byteRecebido[countBytes] = Serial.read();
    sprintf(&str[countBytes], "%c", byteRecebido[countBytes]);
    countBytes++;
    delay(DELAY_TIME_COM);
}

if(countBytes){
    //Serial.print(str);
    //delay(DELAY_TIME_COM);
    descripta_trama(str);
}

if(countBytes && String(receptora) == n_luminaria && valida_CRC(receptora,emissora,codigo,valor,CRC)){

    if(String(codigo) == "0")// Comandos AT
    {
        codigo0();
        Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
        delay(DELAY_TIME_COM);
    }
    else if(String(codigo) == "1")// Stay alive

```

```
{
  codigo1();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "2")// Leitura do sensor de luz
{
  codigo2();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "3")// Leitura do sensor de temperatura
{
  codigo3();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "4")// Leitura do tractus sensor
{
  codigo4();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "5")// Leitura do PWM
{
  codigo5();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "6")// Escrita de PWM
{
  codigo6();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "7")// Escrita no valor maximo de SUN
{
  codigo7();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "8")// Escrita no valor maximo de Temperatura
{
  codigo8();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "9")// Leitura do valor maximo de SUN
{
  codigo9();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "10")// Leitura do valor maximo de Temperatura
{
  codigo10();
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "11")// Escrita em memoria das luminarias circundantes
{
  codigo11();
}
else if(String(codigo) == "12")// Leitura da memoria das luminarias circundantes
{
  codigo12();
}
else if(String(codigo) == "13")//Forcar valor de SUN
{
  luzsensorValue = atoi(valor);
  sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
  Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
  delay(DELAY_TIME_COM);
}
else if(String(codigo) == "14")// Forcar valor de TEMP
{

```

```

tempSensorValue = atoi(valor);
sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
delay(DELAY_TIME_COM);
}
else if(String(codigo) == "15")// Leitura do MODO
{
troca_array(emissora, receptora);
sprintf(valor,"%d",MODE);
sprintf(CRC,"%d",cria_CRC(receptora,emissora,codigo,valor));
Serial.print(cria_trama(receptora,emissora,codigo,valor,CRC));
delay(DELAY_TIME_COM);
}
else if(String(codigo) == "16")// Activação do TRACTUS_AUX
{
for(int k = 10; k < MAX_LUMINARIAS_CIRCUNDANTES +10; k++){
if(EEPROM.read(k)== atoi(emissora) && MODE == SUN){
//MODE = TRACTUS_AUX;
for(int k = 0; k < MAX_LUMINARIAS_CIRCUNDANTES; k++){
if(luminarias_gestao[k][0] == atoi(emissora)){
luminarias_gestao[k][1] = 1;
MODE = TRACTUS_AUX;
}
}
}
}
}
else if(String(codigo) == "17")// Desactivação do TRACTUS_AUX
{
int count_tractus = 0;

for(int k = 10; k < MAX_LUMINARIAS_CIRCUNDANTES + 10; k++){
if(EEPROM.read(k)== atoi(emissora) && MODE == TRACTUS_AUX){
//MODE = SUN;
for(int k = 0; k < MAX_LUMINARIAS_CIRCUNDANTES; k++){
if(luminarias_gestao[k][0] == atoi(emissora)){
luminarias_gestao[k][1] = 0;
}
}
}
}
for(int k = 0; k < MAX_LUMINARIAS_CIRCUNDANTES; k++){
if(luminarias_gestao[k][1] == 1){
count_tractus++;
}
}
if(count_tractus == 0){
MODE = SUN;
}
}
}

//Limpeza das variaveis de comunicação

for(int k=0;k<countBytes;k++){
byteRecebido[k] = 0;
}
for(int k=0;k<MAX_CHAR_VAL;k++){
//_receptora[k] = 0;
}

countBytes = 0;
}

```